PanaXSeries

The One to Watch for Constant Innovation-Making the Future Come Alive

MICROCOMPUTER   MN1030

MN103002A
LSI User's Manual

Panasonic

Request for your special attention and precautions in using the technical informaition and semiconductors described in this book

(1)   An export permit needs to be obtained from the competent authorities of the Japanese Government if any of the products or technologies described in this book and controlled under the "Foreign Exchange and Foreign Trade Law" is to be exported or taken out of Japan.

(2)   The contents of this book are subject to change without notice in matters of improved function.When finalizing your design, therefore, ask for the most up-to-date version in advance in order to check for any changes.

(3)   We are not liable for any damage arising out of the use of the contents of this book, or for any infringement of patents or any other rights owned by a third party.

(4)   No part of this book may be reprinted or reproduced by any means without written permission from our company.

(5)   This book deals with standard specification. Ask for the latest individual Product Standards or Specifications in advance for more detailsd infomation required for your design, purchasing and applications.

> If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.

Table of Contents/List of Figures and Tables

Chapter 13. Electrical Characteristics

Appendices

13

Tabel of Contents / List of Figures and Tables

0

Table of Contents

Chapter 3.   Clock Generator

Chapter 4.   Caches

Chapter 5. Bus Controller

Chapter 7.   Interrupt Controller

Chapter 8.   8-bit Timers

Chapter 9.   16-bit Timers

Chapter 10. Watchdog Timer

Chapter 11. Serial Interfaces

Chapter 13. Electrical Characteristics

Appendices

List of Figures
and Tables

List of Figures

Chapter 5. Bus Controller

Chapter 6.   DMA Controller

Chapter 7.   Interrupt Controller

Chapter 8.   8-bit Timers

Chapter 9.   16-bit Timers

Chapter10. Watchdog Timer

Chapter 11. Serial Interfaces

Chapter 12. I/O Ports

Chapter 13. Electrical Characteristics

Appendices

List of Tables

Chapter 1. General

Chapter 2. CPU

Chapter 3. Clock Generator

Chapter 5. Bus Controller

Chapter 6. DMA Controller

Chapter 7. Interrupt Controller

Chapter 8. 8-bit Timers

Chapter 9. 16-bit Timers

Chapter 10. Watchdog Timer

Chapter 11. Serial Interfaces

Chapter 12. I/O Ports

Chapter 13. Electrical Characteristics

Appendices

Chapter 1.   General

1

1.1 Overview

The MN103002A is a 32-bit microcontroller, geared towards the development of C programming development that offers ease of use and excellent cost-performance, with a simple and high-performance architecture.

Built around a compact 32-bit CPU core with a basic instruction word length of one byte, this microcontroller includes an instruction cache, data cache, bus control circuit, interrupt control circuit, timers, serial interfaces, DMAC, A/D converter, and I/O ports in a 160-pin QFP. This microcontroller is ideal for multimedia devices, which must be able to process large volumes of data (for audio, stills, video, etc.), as well as for real-time control equipment that requires fast and precise control. When supplied with voltage of 3.3 V ± 0.165 V, the MN103002A operates at 66 MHz and achieves performance of 66 MIPS.

MN103002AYB is a product with MN103002A package changed to CSP, and includes the same function and performance as MN103002A.

1.2 Features

Low voltage, high speed processing, and low power consumption (TYP)
- Minimum instruction execution time
    15 ns (when supplied with 3.3 V, and operating with an internal clock speed of 66.6 MHz)
- Power consumption
    500 mW (when supplied with 3.3 V, and operating with an internal clock speed of 66.6 MHz)

Compact and high-performance CPU core
- Simple and highly efficient instruction set
    (Number of basic instructions: 46; number of extension instructions: 7; number of addressing modes: 6)
- Excellent coding efficiency with instructions that have a basic word length of one byte
- Supplied with Sophisticated/high-performance extension operation unit.
    (Supports 30 extension instructions, including a fast multiplication instruction, a sum of products operation instruction, and a saturation operation instruction.)
- Load/store architecture with 5-stage pipeline organization provides fast instruction execution.
- Unique high-speed branch processing
- Supports linear address space of up to 4 GB

Cache memory
- Instruction cache
- Size: 4 Kbytes (2 Kbytes x 2)
- 128 entries; lines size: 16 bytes; 2-way set associative
- LRU replacement algorithm
- Individual ways can be used as RAM (Three possibilities: two-way cache, one-way cache + RAM, or RAM)

● Data cache
  • Size: 4 Kbytes (2 Kbytes x 2)
  • 128 entries, 16-byte line size, two-way set associative
  • Writing policy: Switchable between "write-back" and "write-through"
  • Conversion to RAM in "way" units possible (Three possibilities: 2-way cache, 1-way cache + RAM, and RAM)

Flexible clock control

● Self-excited/externally excited oscillation
  • Clock is supplied either by connecting an oscillator or by inputting a clock signal.
  • Either of the 2 x or 4 x oscillated signal or input clock can be selected for the clock (CPU clock). (Internal maximum: 66.6 MHz)
  (The external bus clock is identical to the oscillation/input clock.)
● Low power consumption mode
  • Supports three modes: HALT, STOP and SLEEP mode.

High-speed/high-performance bus interface

● High speed control of the internal bus/external bus is possible using the CPU clock.
  • Two modes are supported for the internal I/O bus and the external bus: synchronous mode (synchronized with the external bus clock)/asynchronous mode (synchronized with the CPU clock, not synchronized with the external clock).
  • In asynchronous mode, wait states can be controlled for individual CPU cycles.
● External memory space is managed by partitioning it into eight blocks.
  • Chip select signal output for each block.
  • The bus width for each block can be set to 16/32 bits.
  • Blocks 0 to 5 can be switched between synchronous mode and asynchronous mode.
  • Blocks 2 through 7 can be switched between fixed wait state insertion and handshaking.
  • Blocks 1 to 4 can be used as DRAM space.
● Built-in DRAM direct link interface
  • Supports address multiplexing function (8- to 11-bit shift for the low address can be selected).
  • Supports two types of byte specification methods (CAS or WE)
  • Supports high-speed page mode. (Supports the page mode mix cycle DRAM.)
  • Supports CAS-before-RAS refresh. (The refresh cycle is programmable.)
● Avoids time penalty when performing a store operation in the store buffer (one-step)
  • Supports store operations in internal peripheral circuitry and external devices.
  • When the store buffer is empty, the store operation is completed with no wait states, and the CPU can then continue with subsequent processing.

5 V input interface

● Supports 5 V, TTL-level input interface (excluding the oscillator pins).

Wide variety of on-chip peripheral functions

● Interrupts

• 30 sources

- External interrupts: 9 sources ($\overline{\text{IRQn}}$ (n=7 to 0) x 8, and $\overline{\text{NMIRQ}}$ x 1)
- Internal interrupts: 21 sources (timers: 9; SIO: 6; DMAC: 4; WDT: 1; system error: 1)

● Timers

• Four 8-bit timers (all are down-counters)

- Cascaded connection possible (permits use as 16-/24-/32-bit timer)
- Timer output possible (duty ratio: 1:1)
- Internal clock source or external clock source can be selected
- Can be selected as serial interface clocks

• Two 16-bit timers (2 down-counters)

- Cascaded connection possible (permits use as 32-bit timer)
- Timer output possible (duty ratio: 1:1)
- Internal clock source or external clock source can be selected

• One 16-bit timer (up-counter)

- Internal clock source or external clock source can be selected
- Input capture function (rising edge, falling edge, or both edges can be selected)
- Various PWM generation functions (two internal compare and capture registers)

• One watchdog timer

● Serial interface

• Two UART/synchronous (dual use) channels

• One UART (with CTS control) channel

● DMAC

• Number of channels: 4 channels

• Unit of transfer: 8/16/32 bits

• Maximum transfer count: 65536 transfers

• Initiation sources: External requests, various interrupts, software

• Transfer format: Two bus cycle transfers, one bus cycle transfer

• Transfer modes: One-word transfer, burst transfer, intermittent transfer

• Addressing modes:

- Fixed/increment/decrement can be specified for the source and the destination separately.
- Incrementation/decrementation are performed automatically in accordance with the transfer units.

● Input/output ports: 26 (all dual purpose)

● Package

• MN103002A   : QFP160-P-2828B (28 mm square, 0.65 mm pitch QFP)

• MN103002AYB: FLGA165-C-1111 (11 mm square, 0.8 mm pitch CSP)

1.3 Block Diagram
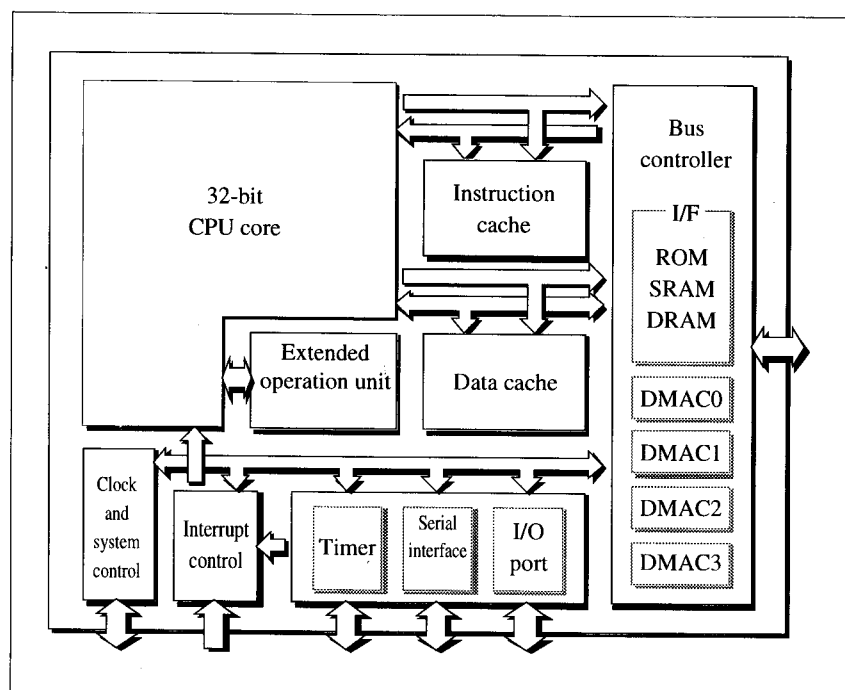
Fig. 1-3-1 MN103002A/MN103002AYB Block Diagram

1.4 Pin Descriptions

1.4.1 Pin Assignments
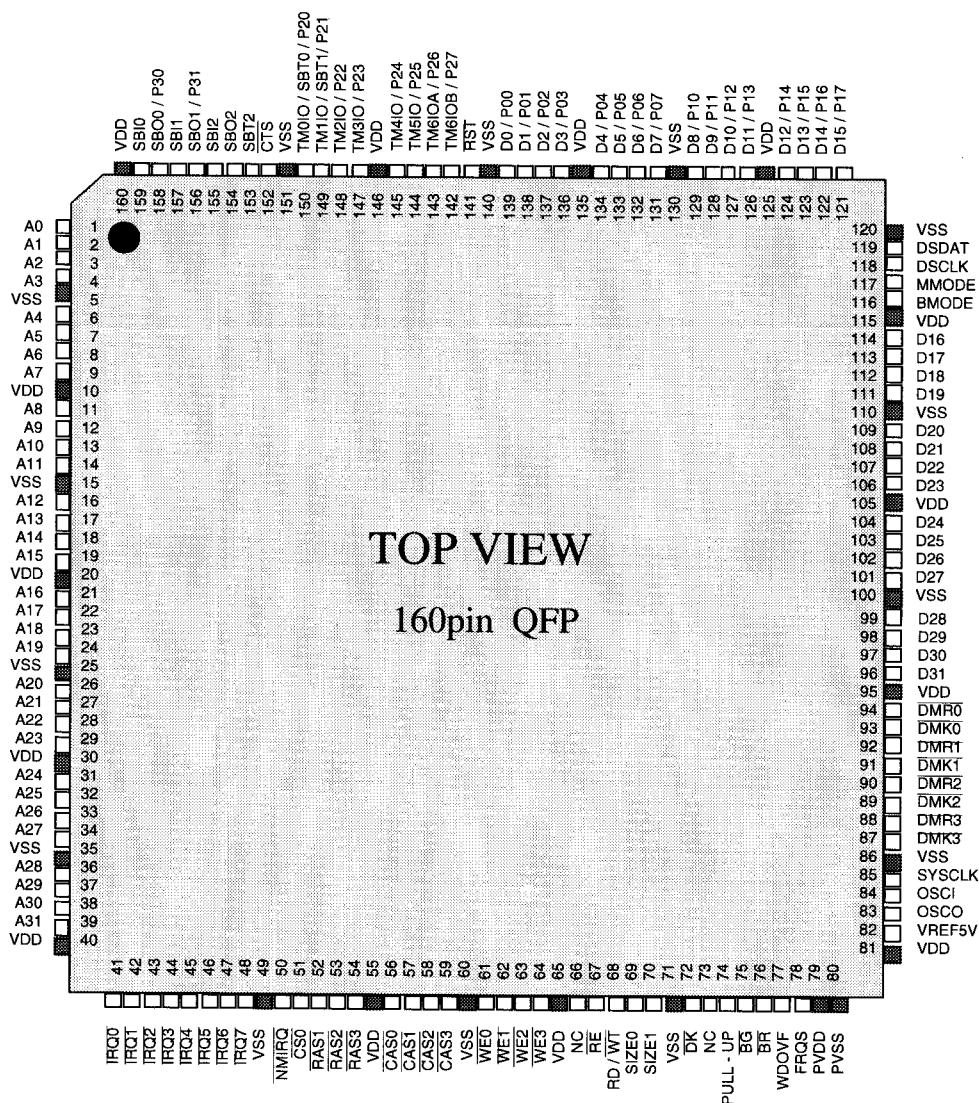
Fig. 1-4-1 MN103002A Pin Assignment Diagram

Table 1-4-1 MN103002A Pin Assignments

No.	Pin name	No.	Pin name	No.	Pin name	No.	Pin name
1	A0	41	IRQ0	81	VDD	121	D15 / P17
2	A1	42	IRQ1	82	VREF5V	122	D14 / P16
3	A2	43	IRQ2	83	OSCO	123	D13 / P15
4	A3	44	IRQ3	84	OSCI	124	D12 / P14
5	VSS	45	IRQ4	85	SYSCLK	125	VDD
6	A4	46	IRQ5	86	VSS	126	D11 / P13
7	A5	47	IRQ6	87	DMK3	127	D10 / P12
8	A6	48	IRQ7	88	DMR3	128	D9 / P11
9	A7	49	VSS	89	DMK2	129	D8 / P10
10	VDD	50	NMIRQ	90	DMR2	130	VSS
11	A8	51	CS0	91	DMK1	131	D7 / P07
12	A9	52	RAS1 / CS1	92	DMR1	132	D6 / P06
13	A10	53	RAS2 / CS2	93	DMK0	133	D5 / P05
14	A11	54	RAS3 / CS3	94	DMR0	134	D4 / P04
15	VSS	55	VDD	95	VDD	135	VDD
16	A12	56	CAS0	96	D31	136	D3 / P03
17	A13	57	CAS1	97	D30	137	D2 / P02
18	A14	58	CAS2	98	D29	138	D1 / P01
19	A15	59	CAS3	99	D28	139	D0 / P00
20	VDD	60	VSS	100	VSS	140	VSS
21	A16	61	WE0	101	D27	141	RST
22	A17	62	WE1	102	D26	142	TM6IOB / P27
23	A18	63	WE2	103	D25	143	TM6IOA / P26
24	A19	64	WE3	104	D24	144	TM5IO / P25
25	VSS	65	VDD	105	VDD	145	TM4IO / P24
26	A20	66	NC *1	106	D23	146	VDD
27	A21	67	RE	107	D22	147	TM3IO / P23
28	A22	68	RD / WT	108	D21	148	TM2IO / P22
29	A23	69	SIZE0	109	D20	149	TM1IO / SBT1 / P21
30	VDD	70	SIZE1	110	VSS	150	TM0IO / SBT0 / P20
31	A24	71	VSS	111	D19	151	VSS
32	A25	72	DK	112	D18	152	CTS
33	A26 / CS4 / RAS4	73	NC *1	113	D17	153	SBT2
34	A27 / CS5	74	PULL - UP *2	114	D16	154	SBO2
35	VSS	75	BG	115	VDD	155	SBI2
36	A28 / CS6	76	BR	116	BMODE	156	SBO1 / P31
37	A29	77	WDOVF	117	MMODE	157	SBI1
38	A30	78	FRQS	118	DSCLK	158	SBO0 / P30
39	A31 / CS7	79	PVDD	119	DSDAT	159	SBI0
40	VDD	80	PVSS	120	VSS	160	VDD

*Those pins for which two or more pin names are defined are dual-purpose pins.

*1) Leave open

*2) Connect a pull-up resistor.

Top View

	13	12	11	10	9	8	7	6	5	4	3	2	1	
N	N.C.	OSCO	DMK3	DMK1	VDD	D28	D25	D22	D19	VDD	N.C.	MMODE	N.C.	**N**
M	WDOVF	VDD	SYSCLK	DMK2	DMK0	D30	D27	VDD	D20	D17	DSDAT	D15/P17	D13/P15	**M**
L	N.C.	PVSS	PVDD	VREF5V	DMR3	DMR1	D31	D21	BMODE	DSCLK	VSS	VDD	D10/P12	**L**
K	BG	N.C.	FRQS	OSCI	VSS	DMR0	VSS	D24	D18	D12/P14	D14/P16	D8/P10	D7/P07	**K**
J	VSS	SIZE0	PULL-UP	DK	BR	DMR2	D26	VSS	D16	D11/P13	D9/P11	D5/P05	VDD	**J**
H	RE	VDD	SIZE1	N.C.	RD/WT	D29	D23	D4/P04	VSS	D3/P03	D6/P06	D2/P02	D0/P00	**H**
G	WE2	WE0	WE1	CAS2	VSS	WE3	N.C.	VDD	D1/P01	TM6IOB/P27	VSS	RST	TM6IOA/P26	**G**
F	CAS3	CAS1	NMIRQ	RAS3/CS3	RAS1/CS1	CAS0	A19	A12	TM0IO/SBT0/P20	TM5IO/P25	TM2IO/P22	TM4IO/P24	TM3IO/P23	**F**
E	VDD	RAS2/CS2	IRQ5	IRQ7	A28/CS6	A22	VDD	A9	SBO2	CTS	SBO1/P31	TM1IO/SBT1/P21	VSS	**E**
D	CS0	VSS	IRQ1	IRQ3	A25	A20	A14	A11	A6	A3	SBO0/P30	SBT2	SBI2	**D**
C	IRQ6	IRQ4	VDD	A30	A27/CS5	VDD	A17	VDD	A4	A1	VDD	SBI0	N.C.	**C**
B	IRQ2	IRQ0	A31/CS7	A26/CS4/RAS4	A23	VSS	A16	A13	A10	A7	VSS	A0	SBI1	**B**
A	N.C.	A29	N.C.	VSS	A24	A21	A18	A15	VSS	A8	A5	A2		**A**
	13	12	11	10	9	8	7	6	5	4	3	2	1	

*1) Leave the N.C. (Not Connected) open

Fig. 1-4-2 MN103002AYB Pin Assignment Diagram

Table 1-4-2 MN103002AYB Pin Assignments

Pin name	No.	Pin name	No.	Pin name	No.	Pin name	No.
A0	B2	IRQ1	D11	OSCO	N12	D13 / P15	M1
A1	C4	IRQ2	B13	OSCI	K10	D12 / P14	K4
A2	A2	IRQ3	D10	SYSCLK	M11	D11 / P13	J4
A3	D4	IRQ4	C12	DMK3	N11	D10 / P12	L1
A4	C5	IRQ5	E11	DMR3	L9	D9 / P11	J3
A5	A3	IRQ6	C13	DMK2	M10	D8 / P10	K2
A6	D5	IRQ7	E10	DMR2	J8	D7 / P07	K1
A7	B4	NMIRQ	F11	DMK1	N10	D6 / P06	H3
A8	A4	CS0	D13	DMR1	L8	D5 / P05	J2
A9	E6	RAS1 / CS1	F9	DMK0	M9	D4 / P04	H6
A10	B5	RAS2 / CS2	E12	DMR0	K8	D3 / P03	H4
A11	D6	RAS3 / CS3	F10	D31	L7	D2 / P02	H2
A12	F6	CAS0	F8	D30	M8	D1 / P01	G5
A13	B6	CAS1	F12	D29	H8	D0 / P00	H1
A14	D7	CAS2	G10	D28	N8	RST	G2
A15	A6	CAS3	F13	D27	M7	TM6IOB / P27	G4
A16	B7	WE0	G12	D26	J7	TM6IOA / P26	G1
A17	C7	WE1	G11	D25	N7	TM5IO / P25	F4
A18	A7	WE2	G13	D24	K6	TM4IO / P24	F2
A19	F7	WE3	G8	D23	H7	TM3IO / P23	F1
A20	D8	RE	H13	D22	N6	TM2IO / P22	F3
A21	A8	RD / WT	H9	D21	L6	TM1IO / SBT1 / P21	E2
A22	E8	SIZE0	J12	D20	M5	TM0IO / SBT0 / P20	F5
A23	B9	SIZE1	H11	D19	N5	CTS	E4
A24	A9	DK	J10	D18	K5	SBT2	D2
A25	D9	PULL-UP [*1]	J11	D17	M4	SBO2	E5
A26 / CS4 / RAS4	B10	BG	K13	D16	J5	SBI2	D1
A27 / CS5	C9	BR	J9	BMODE	L5	SBO1 / P31	E3
A28 / CS6	E9	WDOVF	M13	MMODE	N2	SBI1	B1
A29	A12	FRQS	K11	DSCLK	L4	SBO0 / P30	D3
A30	C10	PVDD	L11	DSDAT	M3	SBI0	C2
A31 / CS7	B11	PVSS	L12	D15 / P17	M2		
IRQ0	B12	VREF5V	L10	D14 / P16	K3		

Pin name	No.
VDD	C3, C6, C8, C11, E7, E13, G6, H12, J1, L2, M6, M12, N4, N9
VSS	A5, A10, B3, B8, D12, E1, G3, G9, H5, J6, J13, K7, K9, L3

*Those pins for which two or more pin names are defined are dual-purpose pins.

[*1] Connect a pull-up resistor.

1.4.2 Pin Functions

Table 1-4-3 List of MN103002A/MN103002AYB Pin Functions

Category	Pin name	I/O	Number of pins	Pin function
Power/ground pins	VDD	I	14	Digital system power supply
	VSS	I	14	Digital system GND
	VREF5V	I	1	5 V reference voltage input
	PVDD	I	1	PLL circuit power supply
	PVSS	I	1	PLL circuit ground
Clock	OSCI	I	1	Oscillator input
	OSCO	O	1	Oscillator output
	SYSCLK	O	1	System clock output
	FRQS	I	1	Input clock frequency switching
Reset	RST	I	1	Reset input
System control	MMODE	I	1	Mode setting signal
	DSCLK	I/O	1	Debugging serial clock I/O
	DSDAT	I/O	1	Debugging serial data I/O
Address bus	A0 - A31	O	32	Address lines 0 to 31 (A26 through A28 and A31 are dual-use pins with $\overline{CS4}$ through 6 and $\overline{CS7}$)
Data bus	D0 - D31	I/O	32	Data lines 0 to 31 (D0 through D15 are dual-use pins with P00 through P07 and P10 through P17)
Bus control	BMODE	I	1	Block 0 bus mode switching signal
	$\overline{RAS1}$ - $\overline{RAS4}$	O	3	Row address strobe signals ($\overline{RAS4}$ only is a dual-use pin with A26)
	$\overline{CAS0}$ - $\overline{CAS3}$	O	4	Column address strobe signals
	$\overline{WE0}$ - $\overline{WE3}$	O	4	Write enable signals
	$\overline{RE}$	O	1	Read pulse signal
	RD / $\overline{WT}$	O	1	Read/write status signal
	SIZE0 - SIZE1	O	2	Access size information notification signals
	$\overline{CS0}$ - $\overline{CS7}$	O	1	Chip select signals ($\overline{CS1}$ through $\overline{CS7}$ are dual-use pins with other signals)
	$\overline{DK}$	I	1	Data acknowledge signal
	$\overline{BG}$	O	1	Bus grant signal
	$\overline{BR}$	I	1	Bus request signal
Serial interfaces	SBI0 - SBI2	I	3	Serial interface 0 to 2 data inputs
	SBO0 - SBO1	I/O	2	Serial interface 0 to 1 data inputs/outputs (dual use)
	SBO2	O	1	Serial interface 2 output
	SBT0 - SBT1	I/O	0	Serial interface 0 to 2 transfer clock input/ output (dual use)
	SBT2	I	1	Serial interface 2 transfer clock input
	$\overline{CTS}$	I	1	"Clear to Send" signal input (serial interface 2 only)

Category	Pin name	I/O	Number of pins	Pin function
8-bit timers	TM0IO	I/O	1	Event count input/toggle output (dual use)
	TM1IO	I/O	1	Event count input/toggle output (dual use)
	TM2IO	I/O	1	Event count input/toggle output (dual use)
	TM3IO	I/O	1	Event count input/toggle output (dual use)
16-bit timers	TM4IO	I/O	1	Event count input/toggle output (dual use)
	TM5IO	I/O	1	Event count input/toggle output (dual use)
	TM6IOA	I/O	1	Capture input/PWM output
	TM6IOB	I/O	1	Event count, capture input/PWM output
Interrupts	$\overline{\text{NMIRQ}}$	I	1	External non-maskable interrupt signal input
	$\overline{\text{IRQ0-IRQ7}}$	I	8	External interrupt signal inputs
Watchdog	WDOVF	O	1	Watchdog overflow output
DMAC	$\overline{\text{DMR0-DMR3}}$	I	4	Direct memory access transfer request inputs
	$\overline{\text{DMK0-DMK3}}$	O	4	Direct memory access transfer acknowledge outputs
I/O ports	P00-P07	I/O	-	Port 0: 8-bit I/O port (dual use with D0 through D7)
	P10-P17	I/O	-	Port 1: 8-bit I/O port (dual use with D8 through D15)
	P20-P27	I/O	-	Port 2: 8-bit I/O port (dual use with TM0IO through TM6IOA and B)
	P30,P31	I/O	-	Port 3: 2-bit I/O port (dual use with SBO0 and SBO1)

Chapter 2. CPU

2

2.1 Overview

Structure

● Load/store architecture

- 9 registers

  (Data registers: 32-bit × 4; address registers: 32-bit × 4; SP: 32-bit × 1)

- Other registers

  (PC: 32-bit × 1; PSW: 16-bit × 1; multiply/divide register: 32-bit × 1; branching registers: 32-bit × 2)

Instructions

● Number of instructions:	46
● Number of addressing modes:	6
● Basic instruction word length:	1 byte
● Code assignment:	1 byte to 2 bytes (basic part) + 0 byte to 6 bytes (extension)

Basic performance

● Maximum internal operating frequency:	66.6 MHz (external oscillation: 16.6 MHz)
● Minimum instruction execution cycle:	1 cycle (15 ns)
● Register-to-register operations:	1 cycle
● Load/store:	1 cycle
● Conditional branch:	1 cycle to 3 cycles

Pipeline

● 5-stage (instruction fetch, decode, execute, memory access, write-back)

Address space

● 4 GB

- Unified space for instructions/data

  (However, instructions can not be fetched from the data-only cache space at x'00000000 to x'1FFFFFFF.)

2.2 Block Diagram

The block diagram focusing on the MN103002A/MN103002AYB CPU is shown below.
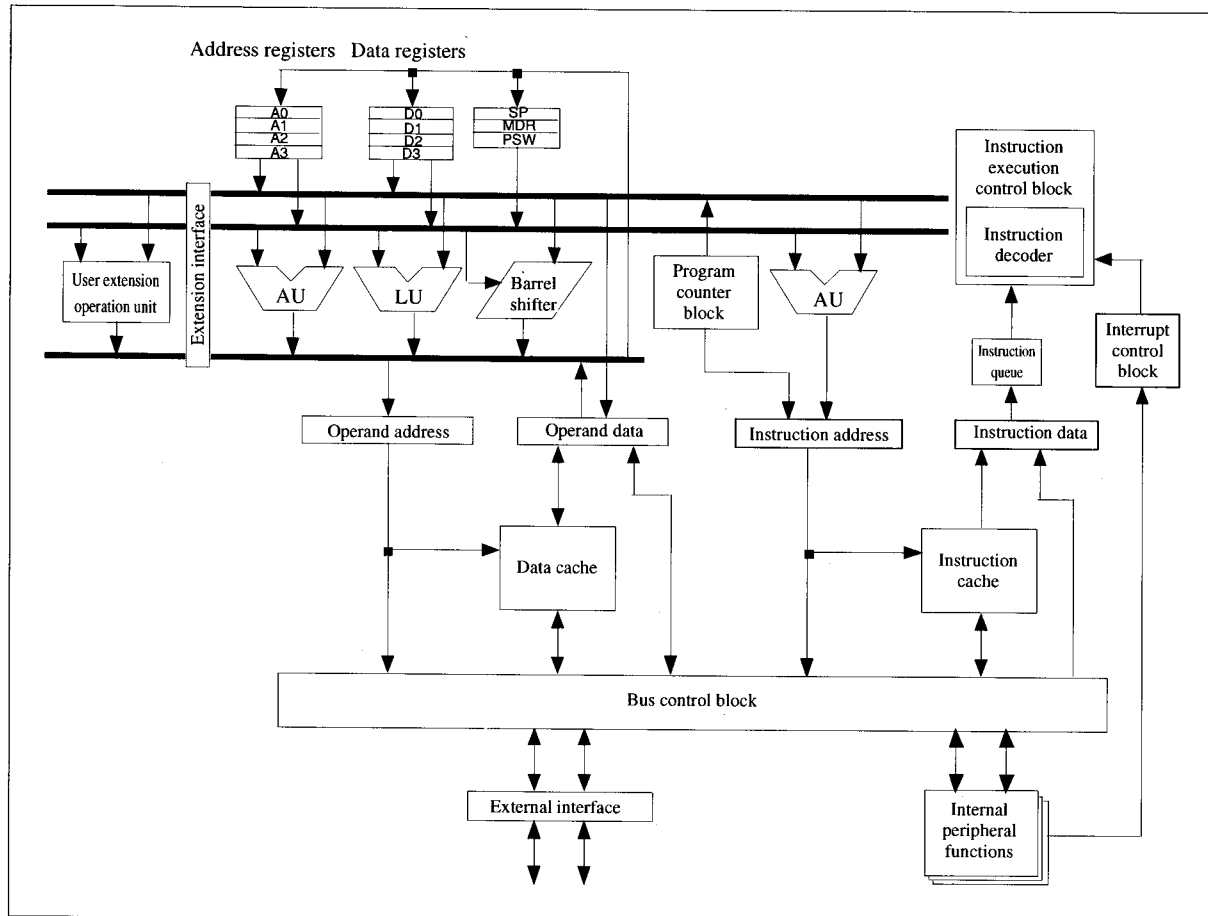
Fig. 2-2-1 CPU Core Block Diagram

2.3   Programming Model

2.3.1   Register Set

The register set is divided into data registers that are used for arithmetic operations, etc., and address registers and a stack pointer that are used for pointers. This arrangement contributes greatly to the improved performance of the internal architecture, through reduction of instruction code sizes, improved parallelism in pipeline processing, etc.

This register set permits programming in C and other high-level languages.
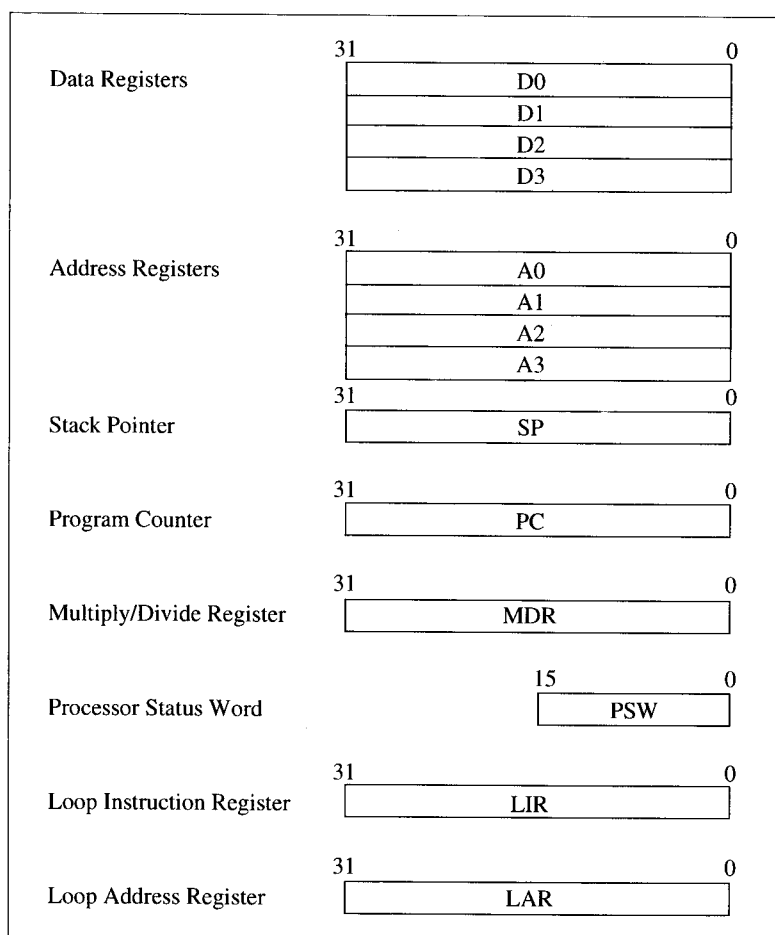
Fig. 2-3-1 Register Set

The Loop Instruction Register (LIR) and the Loop Address Register (LAR) are used for fast execution of branch instructions. Loop control is made faster by using the SETLB instruction to store the branch target instruction and the instruction fetch target address, and then using the Lcc instruction to perform the loop.

● Data Registers (32-bit × 4)

• These are general-purpose registers that can be used for all operations. Operations are performed with 32-bit data; data sizes are converted either when transferring data to or from memory, or by executing the EXTB or EXTH instruction. When 8-bit data is loaded into a data register, it is zero-extended to 32 bits before it is transferred to the register; when storing such data in memory, only the lower 8 bits of the register are transferred to memory. When 8-bit data that is loaded into a data register is to be handled as a signed integer, it is sign-extended from 8 bits to 32 bits by the EXTB instruction. When 16-bit data is loaded into a data register, it is zero-extended to 32 bits before it is transferred to the register; when storing such data in memory, only the lower 16 bits of the register are transferred to memory. When 16-bit data that is loaded into a data register is to be handled as a signed integer, it is sign-extended from 16 bits to 32 bits by the EXTH instruction.

● Address Registers (32-bit × 4)

• These registers are used as address pointers; only instructions for address calculation (addition, subtraction, and comparison) are supported.

• Data in the address registers is used for pointers, and is normally transferred to and from memory with a 32-bit length.

● Stack Pointer (32-bit × 1)

• This pointer points to the top of the stack.

● Program Counter (32-bit × 1)

• This counter points to the address of the instruction that is currently being executed.

● Multiply/Divide Register (32-bit × 1)

• This register is provided for multiply and divide instructions. It holds the upper 32 bits of 64-bit multiplication results for multiply instructions, and the 32-bit remainder for divide instructions. Also, for divide instructions, the upper 32 bits of the dividend are loaded into this register before the division operation is executed.

● Processor Status Word (16-bit × 1)

• This register indicates the status of the CPU. The operation result flags, interrupt mask level, etc., are stored in this register.

* For details on changes in the flag settings, refer to the Instruction Set Manual.

15															0
0	0	S1	S0	IE	IM2	IM1	IM0	0	0	0	0	V	C	N	Z
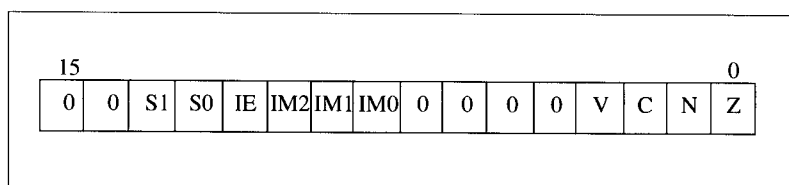
Fig. 2-3-2 Processor Status Word

- Z: Zero Flag

    This flag is set when an operation result is all zeroes, and is cleared by any other result. This flag is also cleared by a reset.

- N: Negative Flag

    This flag is set when the MSB of an operation result is "1", and is cleared if the MSB is "0". This flag is also cleared by a reset.

- C: Carry Flag

    This flag is set when a carry or borrow to or from the MSB is generated in the course of executing an operation, and is cleared if no carry or borrow is generated. This flag is also cleared by a reset.

- V: Overflow Flag

    This flag is set when an overflow occurs in a signed value in the course of executing an operation, and is cleared if no overflow is generated. This flag is also cleared by a reset.

- IM2 to IM0: Interrupt Mask

    These bits indicate the CPU interrupt mask level. The three bits define the mask level from level 0 (000) to level 7 (111), with level 0 being the highest mask level. The CPU accepts only those interrupt requests of a level higher than the mask level indicated here.

    When an interrupt is accepted, the IM bits are set to the priority level of that interrupt. The CPU then does not accept any interrupts of the same interrupt level or lower until the processing of the interrupt that was accepted is completed. The interrupt mask level is set to level 0 (000) by a reset.

- IE: Interrupt Enable

    This bit is normally set to "1", and allows interrupts to be accepted.

    Once the CPU accepts an interrupt request, the IE bit is cleared to "0" and further acceptance of interrupts is prohibited. Therefore, it becomes necessary to set the IE bit again when processing nested interrupts. This bit is cleared by a reset.

- S1 to S0: Software Bits

    These are software control bits that are used by the operating system. This bits cannot be used by general user programs. These bits are cleared by a reset.

● Loop Instruction Register (32-bit × 1)

- This register is provided for the branch instruction (Lcc), and is used by the SETLB instruction to store the branch target instruction. This register works together with the Lcc instruction in order to achieve faster loop control.

● Loop Address Register (32-bit × 1)

- This register is provided for the branch instruction (Lcc), and is used by the SETLB instruction to store the fetch target address.

2.3.2 Control Registers

In the memory mapped I/O method used in the MN103002A/MN103002AYB, the peripheral circuit registers are located in the I/O address space between addresses x'20000000 and x'3FFFFFFF.

The registers listed below are described in this section. For details on other control registers, refer to the chapters that explain the various built-in peripheral functions.

Table 2-3-1 List of Control Registers

Address	Name	Register symbol	Number of bits	Initial value	Access size
x'20000000	Interrupt vector register 0	IVAR0	16	x'XXXX	16
x'20000004	Interrupt vector register 1	IVAR1	16	x'XXXX	16
x'20000008	Interrupt vector register 2	IVAR2	16	x'XXXX	16
x'2000000C	Interrupt vector register 3	IVAR3	16	x'XXXX	16
x'20000010	Interrupt vector register 4	IVAR4	16	x'XXXX	16
x'20000014	Interrupt vector register 5	IVAR5	16	x'XXXX	16
x'20000018	Interrupt vector register 6	IVAR6	16	x'XXXX	16
x'20000040	CPU mode register	CPUM	16	x'0000	16

Interrupt Vector Registers

Register symbol: IVARn

Addresses: x'20000000 (n=0) , x'20000004 (n=1) , x'20000008 (n=2)

x'2000000C (n=3) , x'20000010 (n=4) , x'20000014 (n=5)

x'20000018 (n=6)

Purpose: These registers set the lower 16 bits of the start address of the interrupt handler for each interrupt level.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	IVRNn 15	IVRNn 14	IVRNn 13	IVRNn 12	IVRNn 11	IVRNn 10	IVRNn 9	IVRNn 8	IVRNn 7	IVRNn 6	IVRNn 5	IVRNn 4	IVRNn 3	IVRNn 2	IVRNn 1	IVRNn 0
After reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The interrupt vector registers (IVAR0 to IVAR6) store the lower 16 bits of the start address of the interrupt handler for each interrupt level that is accepted by the CPU. IVAR0 corresponds to level 0 interrupts, and IVAR1 through IVAR6 correspond to level 1 through 6 interrupts, respectively.

Note that the upper 16 bits of the start addresses of the interrupt handlers for each level are fixed at either x'4000 or x'5000.

> **!** For details on how to select the value of the upper 16 bits of the start
> address, refer to section 4.4.1, "Cache Control Register."

CPU Mode Register

Register symbol: CPUM

Address: x'20000040

Purpose: This register sets the clock operation mode for the CPU core and peripheral blocks.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	OSCID	STOP	HALT	SLEEP	OSC1	OSC0
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	OSC0	Always returns "0" when read. Always write "0".
1	OSC1	Always returns "0" when read. Always write "0".
2	SLEEP	CPU operating mode control flag (SLEEP transition request)
3	HALT	CPU operating mode control flag (HALT transition request)
4	STOP	CPU operating mode control flag (STOP transition request)
5	OSCID	Always returns "0" when read.
6 to 15	—	Always returns "0" when read.

The various operating modes can be set by setting the bits as shown in the table below.

Table 2-3-2 Oscillation Control and Operating Mode Control

Operating mode	STOP	HALT	SLEEP	OSC1	OSC0	Clock oscillation	CPU operation clock	Peripheral function operation clock
NOMAL	0	0	0	0	0	Oscillating	Running	Running
HALT	0	1	0	0	0	Oscillating	Stopped	Stopped
SLEEP	0	0	1	0	0	Oscillating	Stopped	Running
STOP	1	0	0	0	0	Stopped	Stopped	Stopped

2.4 Data Formats

This microcontroller can process four data types: bits, bytes, half-words and words. Byte data, half-word data, and word data can each be handled as signed or unsigned data. The most significant bit is the sign bit.

Data in memory must be properly aligned. In other words, the two least significant bits of an address where word data is stored must be "00" (indicating an address that is a multiple of four), and the least significant bit of an address where half-word data is stored must be "0" (indicating an address that is a multiple of two).

The Little Endian format is used for the placement of bytes and bits. Therefore, the address of the byte data on the MSB side of half-word data is the LSB-side byte data address + 1, and the address of the byte data on the MSB side of word data is the LSB-side byte data address + 3. For bit data, the least significant bit is numbered "0", and the bit numbers increase towards the MSB.

Table 2-4-1 Data Types

(1) Bit data

(2) Byte data
    Unsigned 8 bits
    Signed 8 bits   (Sign bit: MSB)

(3) Half-word data
    Unsigned 16 bits
    Signed 16 bits   (Sign bit: MSB)

(4) Word data
    Unsigned 32 bits
    Signed 32 bits   (Sign bit: MSB)

	MSB → LSB			
Bit number	31 24	23 16	15 8	7 0
Address in memory	(Address 4n + 3)	(Address 4n + 2)	(Address 4n + 1)	(Address 4n)
Word data	High-order half word		Low-order half word	
	Most significant byte			Least significant byte
Half-word data			Most significant byte	Least significant byte
Byte data				(byte)
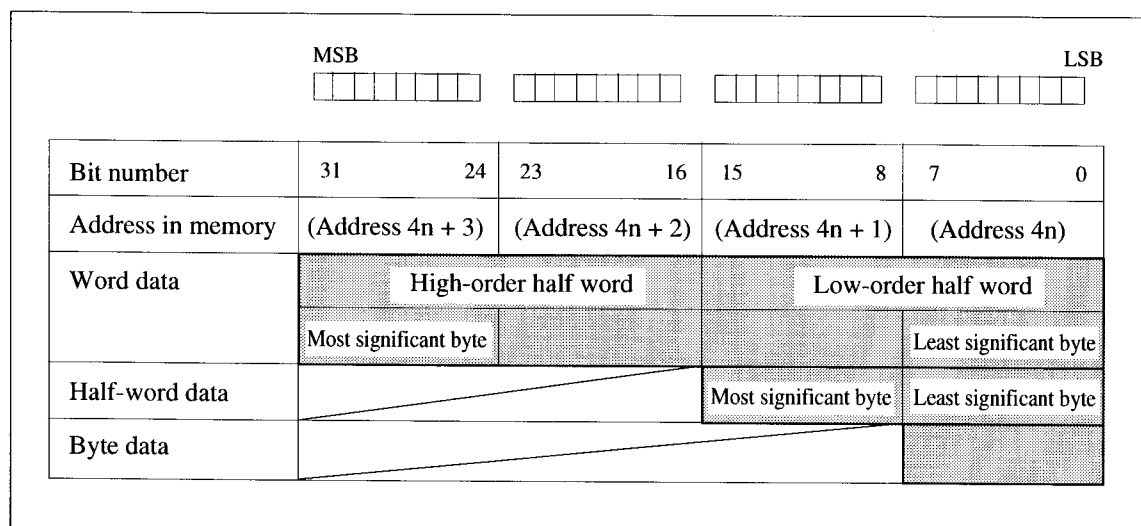
Fig. 2-4-1 Little Endian Format

2.5 Instructions

2.5.1 Instruction Format

There are 10 instruction formats; these are shown below.
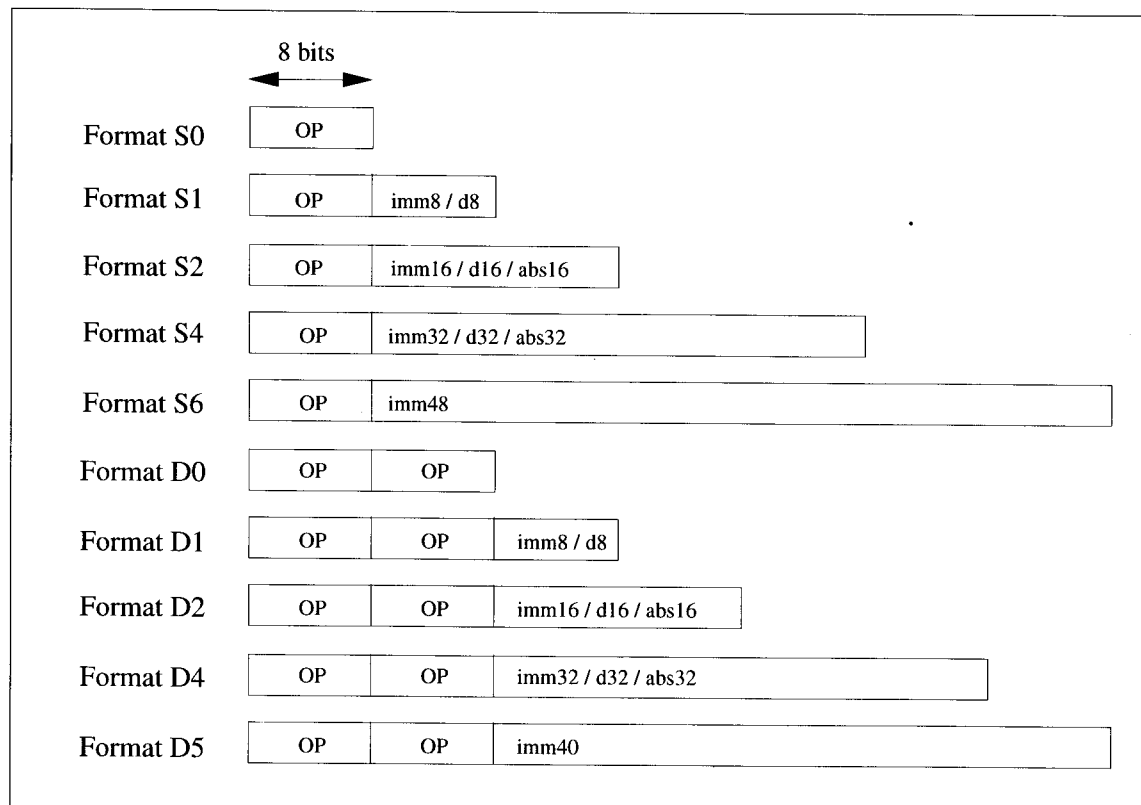
Fig. 2-5-1 Instruction Format Types

Normally, the opcode is followed by an 8-, 16- or 32-bit immediate value, displacement value, or absolute value. However, in instruction formats S2, S4, S6, D2, and D5 above, the opcode is followed by two or more immediate values, displacement values, or absolute values, together representing a 16-bit immediate value (imm16), a 32-bit immediate value (imm32), a 40-bit immediate value (imm40), or a 48-bit immediate value (imm48). Using this notation, the following instructions use 16-, 32-, 40- or 48-bit immediate values.

imm16 :	RET	regs, imm8
	RETF	regs, imm8
	BTST	imm8, (d8,An)
	BSET	imm8, (d8,An)
	BCLR	imm8, (d8,An)
imm32 :	CALL	(d16, PC), regs, imm8
imm40 :	BTST	imm8, (abs32)
	BSET	imm8, (abs32)
	BCLR	imm8, (abs32)
imm48 :	CALL	(d32, PC), regs, imm8

2.5.2 Addressing Mode

This microcontroller supports the 6 addressing modes described below that are frequently used by compilers.

All six addressing modes (register direct, immediate value, register indirect, register indirect with displacement, absolute, and register indirect with index) can be used with data transfer group instructions.

The two addressing modes register direct and immediate addressing can be used with register operation instructions.

Register indirect with index addressing is used to efficiently address arrays and other data.

Table 2-5-1 Addressing Modes

Addressing mode		Address calculation	Effective address
Register direct	Dm / Dn Am / An	—	—
Immediate value	imm8 / regs imm16 imm32 imm40 imm48	—	—
Register indirect	(Am) / (An)	31 0 [Am / An]	31 0 [(32-bit address)]
Register indirect with displacement	(d8, Am) / (d8, An) :d8 is sign-extended (d16, Am) / (d16, An) :d16 is sign-extended (d32, Am) / (d32, An)	31 0 [Am / An] + 31 15 7 0 [d32 / d16 / d8]	31 0 [(32-bit address)] ↑
	(d8, PC) :d8 is sign-extended (d16, PC) :d16 is sign-extended (d32, PC) (Branch instructions only)	31 0 [PC] + 31 15 7 0 [d32 / d16 / d8]	31 0 [(32-bit address)] ↑
	(d8, SP) :d8 is zero-extended (d16, SP) :d16 is zero-extended (d32, SP)	31 0 [SP] + 31 15 7 0 [d32 / d16 / d8]	31 0 [(32-bit address)] ↑
Absolute	(abs16) :abs 16 is zero-extended (abs32)	31 0 [abs32 / abs16]	(32-bit address)
Register indirect with index	(Di, Am) / (Di, An)	31 0 [Am / An] + 31 0 [Di]	31 0 [(32-bit address)] ↑

⚠ **When accessing data in register indirect with displacement and register indirect with index modes, the base address (the contents of Am, An, and SP) and the calculated (effective) address must be located within the same address space.**

For details on memory spaces, refer to section 2.6, "Memory Mode Types and Selection."

2.5.3   Instruction Set

The instruction set has a simple organization, and features the generation of compact and optimized code through a C compiler.

The instruction code size is reduced by making the basic instruction word length one byte. As a result, increases in the size of the assembly language program can be kept to a minimum even though the instruction set is simple, since data transfers to and from memory are limited to load and store operations.

Table 2-5-2 Instruction Types (Total of 46 instructions + extension instructions)

Transfer instructions	Transfers	Sign extension	Clear	
	MOV MOVM MOVBU MOVHU	EXT EXTH EXTB EXTHU EXTBU	CLR	
Arithmetic instructions	Addition	Subtraction	Multiplication	Division
	ADD INC ADDC INC4	SUB SUBC	MUL MULU	DIV DIVU
Compare instruction	Compare			
	CMP			
Logical operation instructions	Logical sum	Logical product	Inversion	Exclusive logical sum
	OR	AND	NOT	XOR
Bit manipulation instructions	Test	Test and set	Test and clear	
	BTST	BSET	BCLR	
Shift instructions	Shift	Rotate		
	ASR ASL LSR ASL2	ROR ROL		
Branch instructions	Branch	Loop setup	Subroutine call	Return
	Bcc Lcc JMP	SETLB	CALL CALLS TRAP	RET RETF RETS RTI
NOP instruction	No operation			
	NOP			
Extension instructions	Extension			
	UDF UDFU			

Note: Interrupts are prohibited and the bus is locked (occupied by the CPU) when BSET or BCLR is being executed. However, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request or DMA transfer may be interposed between the data read and data write by the BSET or BCLR instruction.

If the atomic bus cycles of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken.

1. A program in which a BSET or BCLR instruction is executed should be placed in instruction cache. Note that an external memory access is occurred, if a cache miss occurs in the instruction cache.

2. Program so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

2.5.3.1 Transfer Instructions

Transfer instructions transfer data between registers or between memory and a register. Transfer instructions are grouped in three types: MOV-type instructions, EXT-type instructions, and the CLR instruction. The MOV-type instructions provide data transfer functions, using the various addressing modes. The displacement or the immediate value is sign-extended, depending on the operation. The EXT-type instructions provide the inter-register transfer functions that accompany sign extension. The CLR instruction clears the contents of a register (by loading "0" into it). Except for the CLR function, these functions do not produce changes in flag settings.

Table 2-5-3 List of Transfer Instructions

Instruction	Description
MOV	Transfer of word data between registers
	Transfer of word data between a register and memory
	Transfer of an immediate value into a register
MOVBU	Transfer of byte data between registers and memory (zero extension)
MOVHU	Transfer of half-word data between registers and memory (zero extendsion)
MOVM	Transfer between multiple registers and memory
EXT	64-bit sign extension of word data
EXTB	32-bit sign extension of byte data
EXTBU	32-bit zero extension of byte data
EXTH	32-bit sign extension of half-word data
EXTHU	32-bit zero extension of half-word data
CLR	Data clear

2.5.3.2 Arithmetic Operation Instructions

The arithmetic operation instructions perform arithmetic operations on the source operands and store the results in a register. All of these instructions produce changes in flag settings. The "+1" and "+4" operations, which are used frequently in address calculations, are provided with separate, special instructions.

Table 2-5-4 List of Arithmetic Operation Instructions

Instruction	Description
ADD	Add
ADDC	Add with carry
SUB	Subtract
SUBC	Subtract with borrow
MUL	Signed multiplication
MULU	Unsigned multiplication
DIV	Signed division
DIVU	Unsigned division
INC	Increment by 1
INC4	Increment by 4

2.5.3.3 Compare Instruction

The compare instruction compares the contents of two registers, or the contents of a register with an immediate value. This instruction is used ahead of a conditional branch instruction. The compare instruction produces changes in flag settings.

Table 2-5-5 List of Compare Instruction

Instruction	Description
CMP	Compare

2.5.3.4   Logical Operation Instructions

The logical operation instructions perform logic operations on the source operands and store the results in a register. All of these instructions produce changes in flag settings.

Table 2-5-6 List of Logical Operation Instructions

Instruction	Description
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
NOT	Inversion (one's complement)

2.5.3.5   Bit Manipulation Instructions

The bit manipulation instructions perform bit manipulation operations between an immediate value and the contents of a register, between an immediate value and the contents of memory, or between the contents of a register and the contents of memory. All of these instructions produce changes in flag settings.

Table 2-5-7 List of Bit Manipulation Instructions

Instruction	Description
BTST	Test multiple bits
BSET	Test and set multiple bits (processing unit: byte)
BCLR	Test and clear multiple bits (processing unit: byte)

2.5.3.6　Shift Instructions

The shift instructions shift bits by the specified amount. This operation can be performed within one cycle, regardless of the shift amount. All of these instructions produce changes in flag settings.

Table 2-5-8　List of Shift Instructions

Instruction	Description
ASR	Arithmetic right shift of any number of bits
LSR	Logical right shift of any number of bits
ASL	Arithmetic left shift of any number of bits
ASL2	Arithmetic left shift of two bits
ROR	Rotate right one bit
ROL	Rotate left one bit

2.5.3.7　Branch Instructions

The branch instructions change the flow of program execution according to certain conditions. The conditional branch instructions include normal conditional branch instructions and special loop conditional branch instructions. The special loop conditional branch instructions minimize the branching penalty by using special registers, thus speeding up loop execution. Subroutine calls and returns are advanced functions that manipulate the PC, save and restore multiple registers to and from the stack, and allocate and release stack area.

Table 2-5-9　List of Branch Instructions

Instruction	Description
Bcc	Conditional branch (PC relative)
Lcc	Special loop conditional branch
SETLB	Set start of loop
JMP	Unconditional branch (PC relative, register indirect)
CALL	Subroutine call (advanced type)
CALLS	Subroutine call
RET	Return from subroutine (advanced type)
RETF	Return from subroutine (advanced, fast type)
RETS	Return from subroutine
RTI	Return from interrupt handler
TRAP	Subroutine call to fixed address

2.5.3.8   NOP Instruction

The NOP instruction is an instruction that performs no operation. This instruction is used to wait one cycle without having any effect on any of the system resources.

Table 2-5-10 NOP Instruction

Instruction	Description
NOP	No operation

2.5.3.9   Extension Instructions

The extension instructions are defined for the add-on extension operation unit. The instruction format is predetermined, and the instruction map is also reserved. In the MN103002A/MN103002AYB, 30 extension instructions are implemented, including a high speed multiplication instruction and a sum of products instruction. For details on these instructions, refer to Appendix E, "Extension Operation Instructions."

Table 2-5-11 List of Extension Operations

Instruction	Description
UDF	User extension instruction (sign extension)
UDFU	User extension instruction (zero extension)

2.6 Memory Map

Types of Memory Space

The MN103002A/MN103002AYB has a 4 GB linear address space in which addresses are expressed with 32 bits.

The address space consists of internal I/O space for the chip's internal I/O ports and various control registers, and an external memory space for memory that is external to the chip. The external memory space consists of cacheable space, from which instructions and data that are accessed once are moved to a cache so that they can be subsequently accessed at high speed, and uncacheable space, the contents of which are not moved to a cache when accessed. Furthermore, the cacheable space consists of instruction/data cacheable space in which both instructions and data can be accessed, and data-only cacheable space, in which only data can be accessed.

Instruction strings can be placed in external memory, in either instruction/data cacheable space or uncacheable space. Data can be stored anywhere in memory, and can be referenced via the MOV instruction. Efficient programming is possible because all addressing modes can be used to access data.

When using either register indirect with displacement or register indirect with index addressing mode, the address pointed to by the base register (either Am, An, or SP) (the base address) and the calculated (effective) address must both be in the same address space (either instruction/data cacheable space, data-only cacheable space, uncacheable chip space, or internal I/O space).

Explanation of the Memory Map

The memory map includes a maximum 3 GB of memory spaces (from x'00000000 to x'BFFFFFFF) that can be accessed. The space from x'00000000 to x'1FFFFFFF is one cacheable space (512 MB maximum), a data-only cacheable space in which only data can be accessed. The space from x'20000000 to x'3FFFFFFF is the internal I/O space (512 MB maximum), which is allocated for the chip's internal I/O ports and various control registers. The space from x'40000000 to x'7FFFFFFF is another cacheable space (1 GB maximum), an instruction/data cacheable space in which both instructions and data can be accessed. Finally, the space from x'80000000 to x'BFFFFFFF (1 GB maximum) is uncacheable space.

Note that the unmounted space access is prohibited such as access to the internal I/O space without a control register. The operation is not assured.
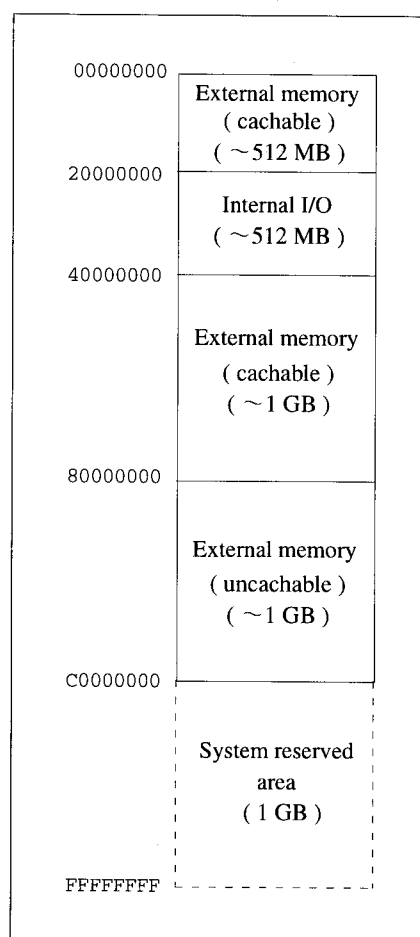
Fig. 2-6-1 MN103002A/MN103002AYB Memory Map

2.7 Interrupt Functions

2.7.1 Overview

The most important key to real-time control is the ability to shift quickly to interrupt handler processing. If an interrupt is generated during the execution of an instruction that requires multiple cycles for execution (multiplication or division instructions, for example), interrupt response is improved by aborting the execution of the instruction and immediately accepting the interrupt. After control returns from the interrupt processing program, the aborted instruction is re-executed.

The speed of interrupt processing and the flexibility of software control are both improved by minimizing the system resources, saved to memory when an interrupt occurs, to just the six bytes of the PC and the PSW.

Fast response and optimal program allocation are also made possible by placing interrupt processing programs (interrupt handlers) at varying addresses for each interrupt level.

The MN103002A/MN103002AYB has the interrupts shown below. When any of these interrupts is generated, control shifts to the appropriate interrupt handler, depending on the source of the interrupt.

Reset interrupt	Highest priority ranking
Non-maskable interrupts	:
Level interrupt n (n = 0 to 6)	Lowest priority ranking

Fig. 2-7-1 shows an overview of the interrupt system. The MN103002A/MN103002AYB has 28 interrupt group control blocks external to the CPU, and controls the interrupts of each group separately. Each interrupt group control block can accept up to three interrupt requests; the controller as a whole can handle up to 30 interrupt sources.

Except for reset interrupts, interrupts from the timer and other peripheral circuits and external pin interrupts are registered in all of the interrupt group control blocks. Those interrupt requests that pass the interrupt mask level (a level from 0 to 6) that is set in the interrupt group control blocks are output to the CPU. Group 0 is allocated for nonmaskable interrupts only.
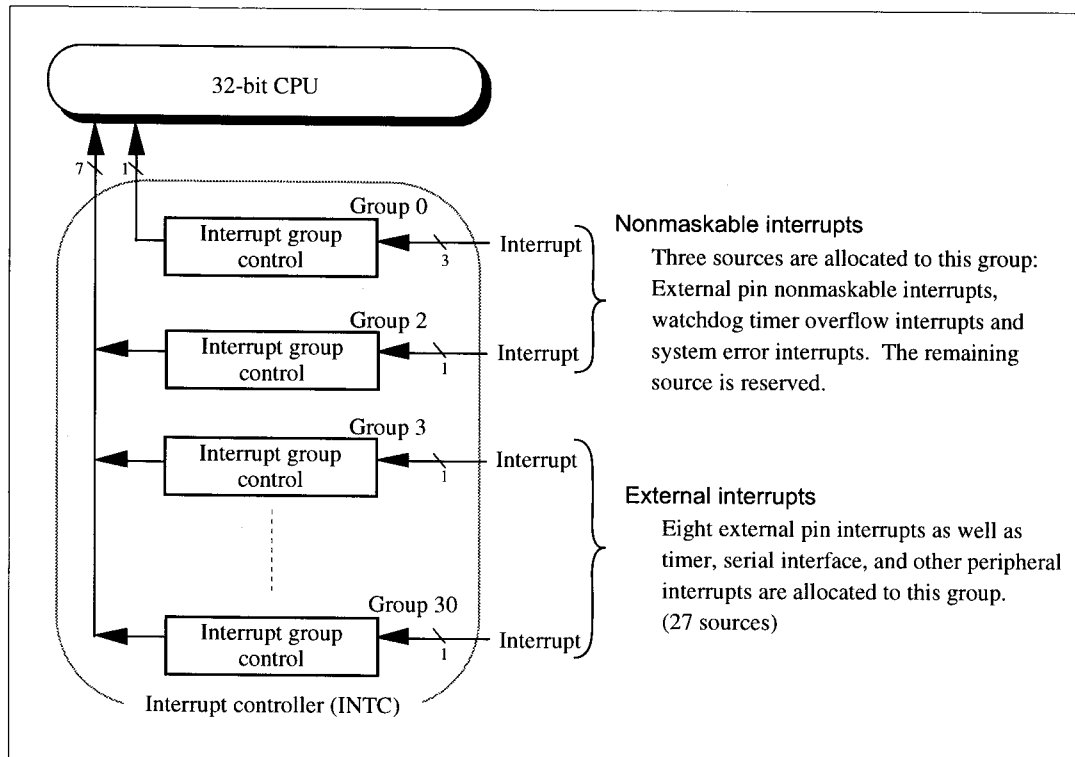
Fig. 2-7-1 Overview of the Interrupt System

2.7.2　Interrupt-related Registers

2.7.2.1　Processor Status Word Register (PSW)

Interrupt Enable and Interrupt Mask Level are the two interrupt-related flags in the Processor Status Word register (PSW). These flags can be both read and written. For details on the PSW, refer to section 2.3.1, "Register Set."

IE (Interrupt Enable)

This flag enables all interrupts except for nonmaskable interrupts and reset interrupts. Interrupts are enabled when IE = 1. When the system is reset, IE = 0.

Once an interrupt is accepted, IE is cleared (interrupts are prohibited). Set IE when accepting nested interrupts within an interrupt handler.

IM2 to IM0 (Interrupt Mask)

These bits hold the current interrupt mask level. When IE = 1, the CPU accepts interrupts of a level higher than IM. Level 0 (000) is set when the system is reset or started up.

The following table shows the relationship between the mask level and the acceptable interrupt levels.

Table 2-7-1　Relationship between Mask Levels and Acceptable Interrupt Levels

Interrupt mask level			Interrupt levels that can be accepted
IM2	IM1	IM0	
0	0	0	Interrupts disabled (only non-maskable interrupts accepted)
0	0	1	0
0	1	0	0∼1
0	1	1	0∼2
1	0	0	0∼3
1	0	1	0∼4
1	1	0	0∼5
1	1	1	0∼6

2.7.2.2 Interrupt Control Registers (GnICR)

The Interrupt Control Registers (GnICR: n = 0, 2 to 10, 12 to 21, and 23 to 30), which control peripheral interrupts external to the CPU, actually combine an interrupt priority level register, an interrupt enable register, an interrupt request register, and an interrupt detection register into a single register. There are 28 Interrupt Control Registers, one for each group, and they are located in the control register space from x'34000100 to x'34000179. G0ICR is dedicated for non-maskable interrupts, so G0ICR is also called NMICR (from the least significant bit: external pin non-maskable interrupt, watchdog overflow interrupt, system error interrupt).

Fig. 2-7-2 shows the Interrupt Control Register (GnICR) configuration. Each field is explained in detail below.
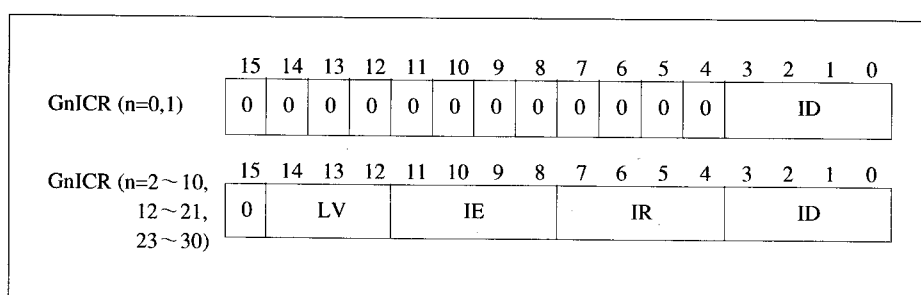
Fig. 2-7-2 Interrupt Control Register (GnICR)

LV2 to LV0 (Interrupt Priority Level Register)

This 3-bit field sets the interrupt priority level. When the interrupt level set in LV2 to LV0 is smaller than IM2 to IM0 in the PSW, interrupts in the corresponding interrupt group are enabled. All interrupts (maximum of four) in the same interrupt group have the interrupt level specified by LV2 to LV0.

When interrupt requests are generated simultaneously from multiple interrupt groups, those with the highest priority level are accepted. Furthermore, when multiple interrupt groups have the same interrupt priority level, interrupts from the group with the highest priority ranking (the interrupt group with the smallest group number) are accepted.

All bits are cleared to "0" when the system is reset.

IE3 to IE0 (Interrupt Enable Register)

This field consists of a maximum of four bits and specifies whether individual interrupts are enabled or not. Each bit from IE3 to IE0 corresponds to one of the interrupt sources (maximum of four) in the interrupt group. When any of the bits from IE3 to IE0 are set to "1", the corresponding interrupts are enabled.

An interrupt is generated when any bit from IR3 to IR0 and the corresponding bit from IE3 to IE0 are set.

All bits are cleared to "0" when the system is reset.

IR3 to IR0 (Interrupt Request)

This field consists of a maximum of four bits and registers interrupt requests. Each bit from IR3 to IR0 corresponds to an interrupt. Once an interrupt is accepted, the interrupt handler is responsible for clearing the corresponding bit from IR3 to IR0.

All bits are cleared to "0" when the system is reset.

The conditions for setting and clearing IR3 to IR0 are listed below.

ID3 to ID0 (Interrupt Detect Register)

This field consists of a maximum of four bits and stores the logical product of each bit from IE3 to IE0 and with each bit from IR3 to IR0. If an interrupt that is enabled by IE3 to IE0 is generated, the ID bit corresponding to that interrupt goes to "1". This field is used to detect specific interrupts within a group during interrupt processing.

Interrupt requests are cancelled by writing specific values in IR3 to IR0 and ID3 to ID0, and clearing the interrupt request field.

Table 2-7-2 Changes in IR (GnICR: n = 0)

Write IR	IR after write
0	Unchanged
1	0

Table 2-7-3 Changes in IR (GnICR: n = 2 to 10, 12 to 21, or 23 to 30)

Write IR	ID	IR after write
0	0	Unchanged
0	0	0
1	0	Unchanged
1	1	1

2.7.2.3 Interrupt Accept Group Register (IAGR)

The Interrupt Accept Group Register (IAGR) stores the current lowest interrupt group number amang the group numbers of the interrupts that have been accepted by the CPU with the same interrupt level indicate by IM2 to IM0 in the PSW. This register is located at address x'34000200 in the control register space. The GN4 to GN0 field (5 bits) corresponds to the interrupt group number.

One way to determine the branch destination of the interrupt program for an individual group is by referencing the contents of the address that is the sum of the starting address of the interrupt vector table and the value in the Interrupt Accept Group Register. The Interrupt Accept Group Register is a read-only register, and cannot be written. If an interrupt source of the appropriate interrupt level does not exist, "0" is returned when IAGR is read.

Accessing the IAGR has no meaning during non-maskable interrupts.
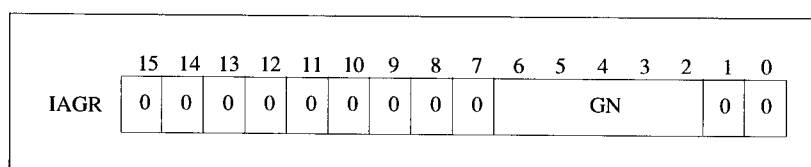
Fig. 2-7-3   Interrupt Accept Group Register

2.7.2.4 Interrupt Vector Registers

The Interrupt Vector Registers (IVAR0 to IVAR6) store the lower 16 bits of the start address of the interrupt handler for interrupts of the accepted level. These registers are located at addresses x'20000000 to x'2000001B in the control register space. IVAR0 to IVAR6 store the start addresses corresponding to interrupt levels 0 through 6. When an interrupt is generated, control is transferred to the 32-bit address of which the upper 16 bits are either x'4000 or x'5000 and the lower 16 bits are the value stored in the IVARn register that corresponds to the level of the interrupt that was generated. These registers are undefined when the system is reset.
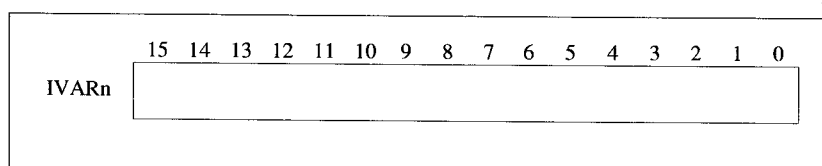
Fig. 2-7-4   Interrupt Vector Register

! For details on how to select the upper 16 bits of the start address, refer to section 4.4.1, "Cache Control Register."

2.7.3 Interrupt Types

2.7.3.1 Reset Interrupt

The reset interrupt has the highest priority level, and is generated by setting the $\overline{\text{RST}}$ pin low. A reset interrupt causes the registers, etc., to be initialized. If the reset pin goes high, the microcontroller waits until the oscillation of the internal clock stabilizes, and then begins executing program instructions starting from address x'40000000.

2.7.3.2 Non-maskable Interrupts

Nonmaskable interrupts are accepted regardless of the values of the interrupt enable (IE) flag and the interrupt mask (IM2 to IM0) in the PSW. Non-maskable interrupt sources are the external pin non-maskable interrupt source, the watchdog timer overflow interrupt source, and the system error interrupt source.

When a nonmaskable interrupt is accepted, control transfers to an interrupt handler that starts at address x'40000008.

The interrupt handler accesses NMICR to analyze the source of the interrupt, performs the interrupt processing, cancels the interrupt source, and then returns to the normal program using the RTI instruction.

External pin non-maskable interrupts

An external nonmaskable interrupt is generated when the $\overline{\text{NMIRQ}}$ pin goes low. If an external nonmaskable interrupt is generated, the external non-maskable interrupt request flag (NMIF) in the Non-maskable Interrupt Control Register (NMICR) is set to "1".

Watchdog timer overflow interrupts

A watchdog overflow interrupt is generated when the watchdog timer enable flag (WDCNE) in the Watchdog Timer Control Register (WDCTR) is "1" and a watchdog timer overflow occurs. If a watchdog interrupt is generated, the watchdog overflow interrupt request flag (WDIF) in the Nonmaskable Interrupt Control Register (NMICR) is set to "1".

System error interrupts

System error interrupt occurs when an unaligned memory access or an unimplemented instruction is executed or other fatal error occurs. If a system error interrupt is generated, the system error interrupt request flag (SYSEF) in the Non-maskable Interrupt Control Register (NMICR) is set to "1".

> **!** Do not change the interrupt enable (IE) in PSW during non-maskable interrupt proccessing.

2.7.3.3　Level Interrupts

Level interrupts are interrupts for which the interrupt level can be controlled through the interrupt enable flag (IE) and interrupt mask (IM2 to IM0) bits in the PSW. Level interrupts are interrupts from the interrupt group controllers that are external to the CPU (in other words, peripheral interrupts). There are 27 groups and 27 interrupt sources.

Each interrupt group controller includes an interrupt control register (GnICR); the interrupt priority level can be set independently for each interrupt group. It is also possible to set the same interrupt priority level for different interrupt groups. If interrupts of the same priority level are generated simultaneously, the interrupts are accepted in the order set by the hardware (the interrupt group with the lowest interrupt group number is given priority).

When a maskable interrupt is accepted, control branches to the 32-bit address of which the upper 16 bits are either x'4000 or x'5000 and the lower 16 bits are the value stored in the IVARn register that corresponds to the level of the interrupt that was generated.

The interrupt handler accesses IAGR to analyze the interrupt group, accesses GnICR (n = 2 to 10, 12 to 21, or 23 to 30), analyzes the interrupt source, performs the interrupt processing, cancels the interrupt source, and then returns to the normal program using the RTI instruction.

2.7.4   Interrupt Operation

When the MN103002A/MN103002AYB accepts an interrupt, first the sequence that is processed by the hardware automatically is executed. Control then shifts to the software interrupt handler, and the interrupt handler is started up.

The interrupt processing sequence is described below.

2.7.4.1   Interrupt Sequence

Hardware interrupt processing sequence

Step 1:   The contents of the PSW are saved to the stack. (SP-8).

Step 2:   The PC (the return address) is saved to the stack (SP-4).

Step 3:   The contents of the PSW are updated.

IE is cleared and the level of the interrupt that was accepted is stored in IM2 to IM0. (In the case of non-maskable interrupts, IM2 to IM0 are undefined.)

Step 4:   The contents of the stack pointer are updated. (SP-8 $\rightarrow$ SP)

Step 5:   Control jumps to the fixed address corresponding to the source of the interrupt that was accepted, or to the address indicated by the Interrupt Vector Register (IVARn).

When an interrupt other than a reset interrupt is accepted, control jumps to the fixed address corresponding to the source of the interrupt, or to the address indicated by the Interrupt Vector Register. The processing indicated below is then performed at the jump destination in order to evaluate the interrupt source in further detail.

> **For details on processing in the case of a reset interrupt, refer to section 2.7.3.1, "Reset Interrupt."**
>
> **Tormally, a branch instruction (JMP instruction, etc.) is placed at the Jump destination for the reset interrupt, transferring control to the initialization program.**

Example of preprocessing by the interrupt handler

Step 1: The contents of the registers are saved.

The registers that are saved are those that will be used by the interrupt handler.

Step 2: The interrupt group analysis is performed.

Step 2.1: The interrupt acknowledge sequence is executed.

An "interrupt acknowledge" consists of reading the Interrupt Accept Group Register (IAGR) to obtain the group number of the interrupt group with the highest priority among the specified interrupt levels.

Step 2.2: The starting address of the interrupt handler for the individual level is generated.

Step 2.3: Control is transferred to the interrupt handler for the individual level.

Step 3: If there are multiple sources within a single group, the individual sources are specified by reading the Interrupt Control Register (GnICR).

- In the case of non-maskable interrupts, the source is specified by accessing the G0ICR (NMICR) directly without accessing the IAGR.

Step 4: Control is transferred to the interrupt handler for the individual interrupt source.

Note that because this microcontroller uses a store buffer when writing data via the bus controller, when releasing the interrupt source it is necessary to read the appropriate register immediately after clearing the interrupt source in order to wait for the source in the GnICR to be cleared completely.

Example of post-processing by the interrupt handler

Step 5: The contents of the registers are restored.

The registers that are restored are those that were saved in the preprocessing.

Step 6: The RTI instruction is executed and control returns to the program before the interrupt.

Fig. 2-7-5 shows the flow of the interrupt sequence (when nested interrupts are not accepted).

The numbers shown in this figure correspond to the step numbers of the processing performed by the interrupt handler as described above.
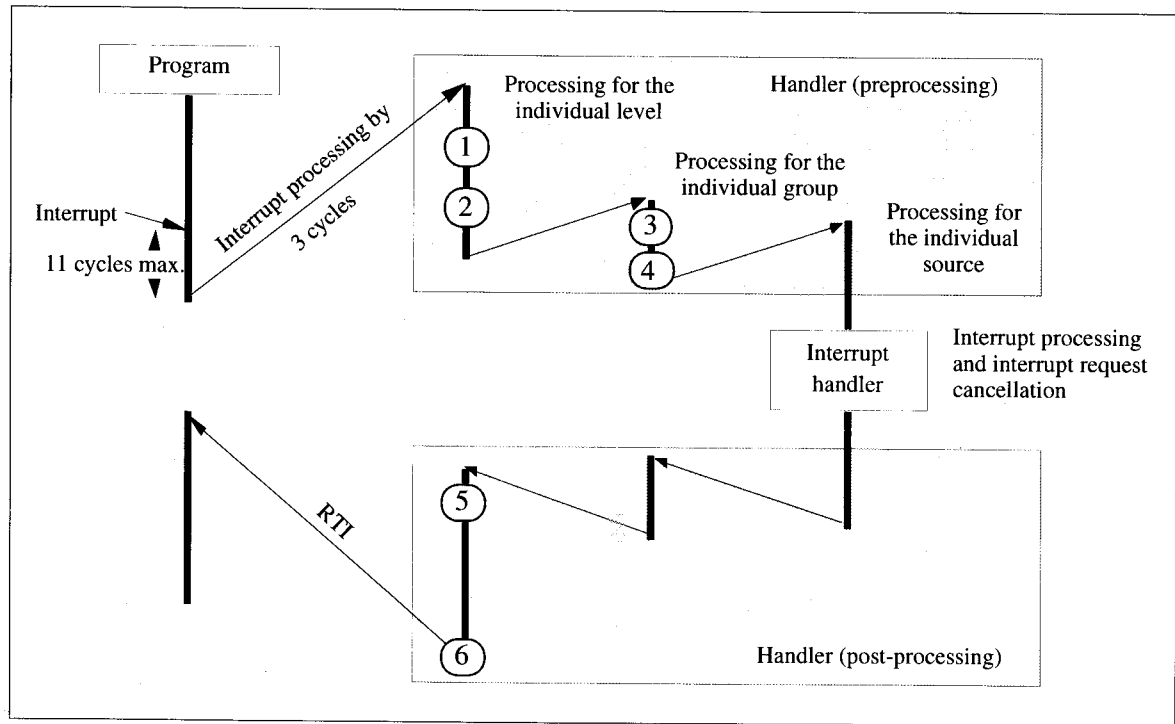
Fig. 2-7-5   Flow of Interrupt Sequence

Even faster interrupt response can be attained by assigning only one source or just a few sources to one interrupt level.

Fig. 2-7-6 shows the flow of the interrupt sequence when only one source was assigned to one interrupt level.
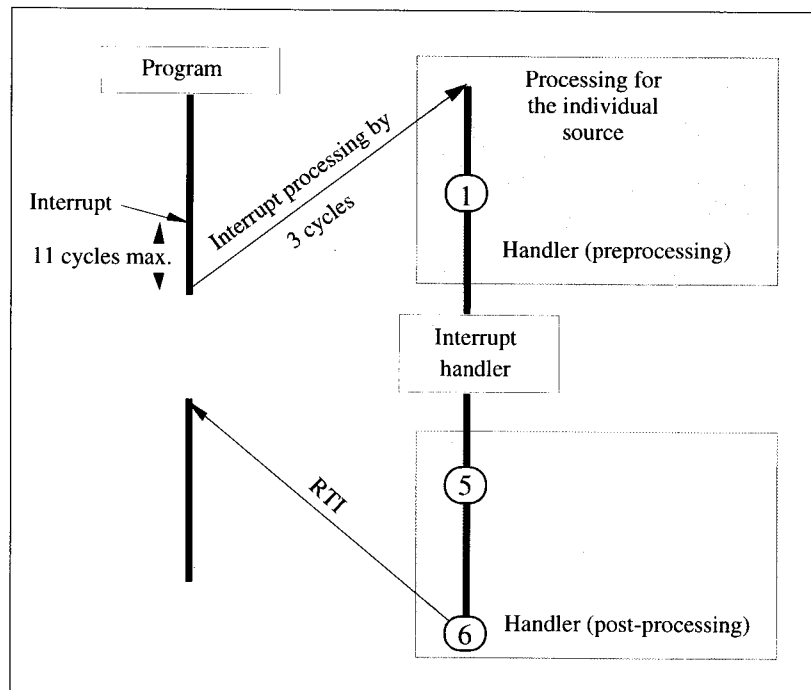
Fig. 2-7-6   Flow of Interrupt Sequence

2.7.4.2　Nested interrupts

When a level interrupt is generated, nested interrupts are prohibited by clearing the IE flag in the PSW. Nested interrupts can be enabled even while in the midst of processing a level interrupt by setting the IE flag to "1". However, in order for a nested interrupt to be generated, the interrupt must still have a higher priority than that indicated by the interrupt mask level bits IM2 to IM0 in the PSW. (In other words, the interrupt level indicated by LV2 to LV0 in the GnICR must be smaller than the interrupt mask level bits IM2 to IM0 in the PSW.)

When a nonmaskable interrupt is generated, nesting of level interrupts and nonmaskable interrupts is prohibited until the nonmaskable interrupt handler terminates and executes the RTI instruction.

2.7.4.3　Interrupt Acceptance Timing

If an interrupt request is generated while an instruction is being executed, execution of the instruction is aborted, if possible (even in the case of an instruction that requires multiple execution cycles, such as a multiple or divide instruction), and the interrupt is accepted. The aborted instruction is executed again after control returns from the interrupt processing. By aborting instructions in this case, the interval during which interrupt acceptance is prohibited is kept to no more than 11 cycles. (The maximum interval of 11 cycles occurs only in special cases, such as task or context switching, in which all of the registers are being saved or restored by an instruction such as MOVM, CALL, or RET.)

2.7.4.4　Stack Frame

When an interrupt is accepted, a stack frame is allocated and a total of six bytes of information, consisting of the PC and the PSW, are saved in the stack in order to return from the interrupt later. However, since the transfer of data that spans a 32-bit boundary is prohibited, the SP value must always be set to a multiple of four. Therefore, as shown in Fig. 2-7-7, the stack frame is allocated so that the value of the SP is always a multiple of four. As a result, the six bytes of information are saved to an eight-byte area.
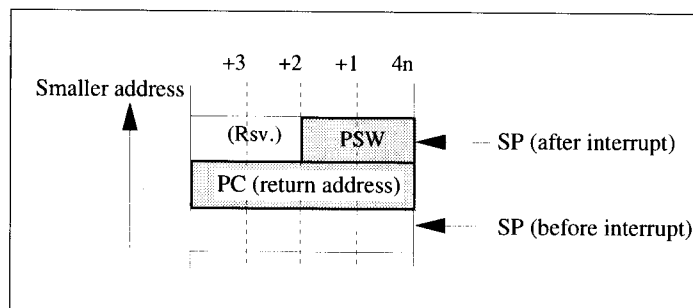
Fig. 2-7-7　Stack Frame Configuration

2.8 Operating Modes

2.8.1 Overview

The MN103002A/MN103002AYB has three basic operating modes. In order to support low power consumption, this microcontroller is equipped with switching control functions that start/stop oscillation and start/stop the CPU and peripheral circuits.

1. Reset mode (RESET)
2. Normal operating mode (NORMAL)
3. Low power consumption modes:
   Stop mode (STOP)
   Halt mode (HALT)
   Sleep mode (SLEEP)

> **!** No matter which state the microcontroller is in, a reset always returns the microcontroller to the normal operating mode.
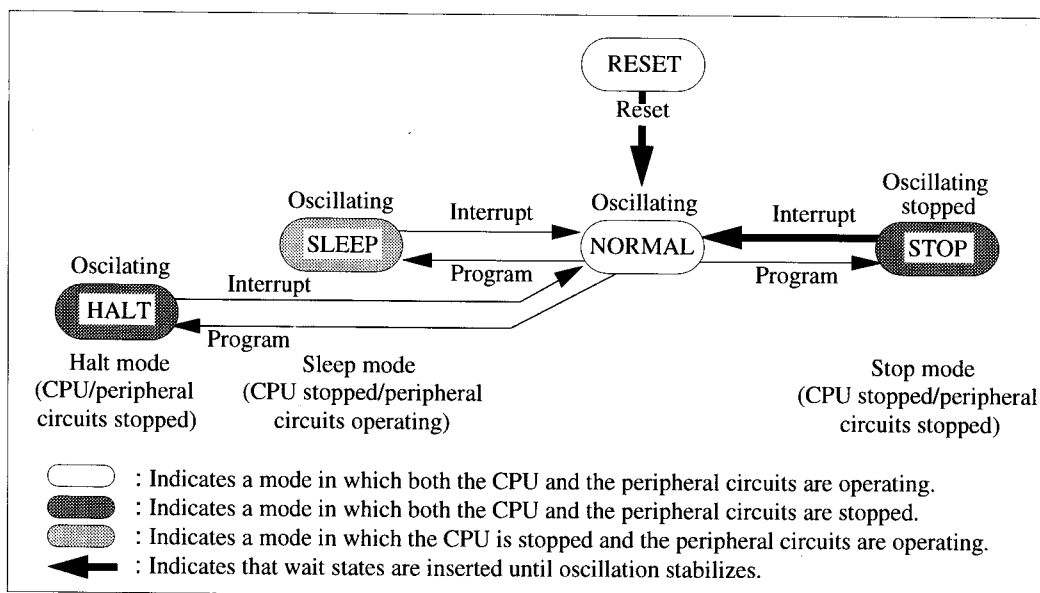
Fig. 2-8-1 Operating Mode Transition Diagram

> **!** Mode transitions are controlled by programs by setting the CPUM register.

2.8.2 Reset Mode

"Reset mode" is the state in which the reset pin is active (low).

If the reset pin goes active, the chip resets (initializes) itself internally and begins waiting for oscillation to stabilize by means of an 18-bit binary counter that is driven by the oscillation clock.

During fast oscillation: Oscillation stabilization wait time for oscillating frequency f_{OSC}:

$$t_{OSCW} = 2^{18} \times 1/f_{OSC}$$

In other words, when $f_{OSC} = 16.5$ MHz:

$$t_{OSCW=66\,MHz} = 15.888 \text{ ms}$$

Table 2-8-1 shows the status of the CPU registers immediately after a reset:

Table 2-8-1   CPU Register Status Immediately after a Reset

Register	Status
PC	x'40000000
D3~D0	Undefined
SP	Undefined
A3~A0	Undefined
MDR	Undefined
LIR	Undefined
LAR	Undefined
PSW	x'0000

After the wait for oscillation stabilization is completed, the internal reset is released and the microcontroller enters normal operation mode.

2.8.3 Low Power Consumption Modes

Low power consumption is attained by halting the oscillation of the oscillator itself or PLL oscillation, and by stopping the clock that is supplied to the CPU and peripheral circuits. The three low power consumption modes are listed below.

STOP mode

In this mode, oscillation of the oscillator itself and PLL oscillation are stopped. When an interrupt is generated, the oscillator and the PLL start operating, the microcontroller waits for the oscillation to stabilize, and then the microcontroller enters the NORMAL operation mode.

HALT mode

In this mode, although the oscillator itself and the PLL are operating, the clock that is supplied to the CPU and peripheral circuits is stopped and the CPU and peripheral circuits stop operating.

When an interrupt is generated, the microcontroller enters the NORMAL operation mode.

SLEEP mode

In this mode, although the oscillator itself and the PLL are operating, the clock that is supplied to the CPU is stopped so that the CPU stops but the peripheral circuits are operating.
When an interrupt is generated, the microcontroller enters the NORMAL operation mode immediately.

Chapter 3.   Clock Generator

3

3.1 Overview

The clock generator (CG) in the MN103002A/MN103002AYB has a built-in PLL circuit, and in addition to sup-plying clock pulses at a frequency that is a multiple of the oscillating frequency of the oscillator, the CG also supplies clock pulses with the same frequency as the oscillating frequency of the oscillator to external devices.

3.2 Features

The features of the CG are described below.

● Flexible clock control
- Supports self-excitation/external excitation (input frequency: 13.0 MHz to 33.3 MHz)
- When the input frequency is 16.6 MHz or less, a clock that is four times the input frequency is supplied as the CPU clock (MCLK); if the input frequency exceeds 26 MHz, a clock that is twice the input frequency is supplied. The maximum frequency for MCLK is 66.6 MHz.
- When the input frequency is 16.6 MHz or less, a clock that has the same frequency as the input frequency is supplied as the peripheral clock (IOCLK); if the input frequency exceeds 26 MHz, a clock that is one-half the input frequency is supplied.
The maximum frequency for IOCLK is 16.6 MHz.

3.3 Block Diagram
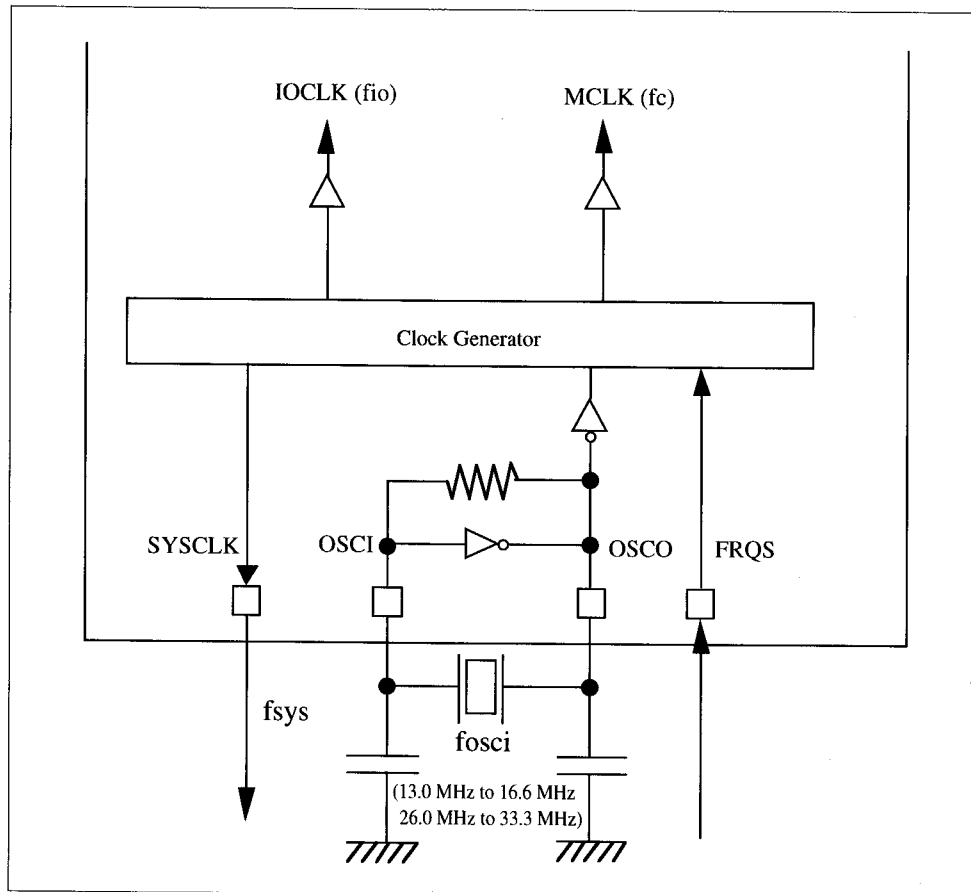
Fig. 3-3-1. Clock Generator

3.4 Description of Operation

3.4.1 Input Frequency Setting

The CG input frequency range is set through the external input pin FRQS. When the input frequency (fosci) is generated using an oscillator or resonator such that 13.0 MHz≤ fosci ≤ 16.6 MHz, set FRQS high. When the input frequency (fosci) is generated using an oscillator or resonator such that 26.0 MHz ≤ fosci ≤ 33.3 MHz, set FRQS low. Using an oscillator or resonator that generates a frequency lower than 13.0 MHz, between 16.6 MHz and 26.0 MHz, or higher than 33.3 MHz is prohibited. The correspondence between the input frequency range and the FRQS mode is shown in Table 3-4-1.

Table 3-4-1 FRQS Modes Versus Input Frequency Ranges

Input frequency range	FRQS mode
13.0 MHz ≤ fosci ≤ 16.6 MHz	1
26.0 MHz ≤ fosci ≤ 33.3 MHz	0

3.4.2 Internal Clock Supply

Depending on the value of the external input pin FRQS, the CPU core/internal RAM/bus controller operating clock (MCLK) frequency is either twice or four times the input frequency, and the internal peripheral function operating clock (IOCLK) frequency is either the same as or one-half the input frequency. Note that the clock (SYSCLK) that is supplied to external devices is the same frequency as the input frequency, with a 50 % duty ratio.

When the reset state is released, SYSCLK, MCLK, and IOCLK are supplied starting after a certain oscillation stabilization wait time. At 16.5 MHz, the oscillation stabilization wait time is 15.888 ms.

Notes on use:
- **Using an oscillator with an input frequency between 16.6 MHz and 26 MHz, lower than 13.0 MHz, or higher than 33.3 MHz is prohibited, because the operation of the PLL circuit cannot be guaranteed.**
- **When using an externally supplied clock, input the clock to the OSCI pin and leave the OSCO pin open.**

Note: When changing the input frequency during operation, be certain to set the FRQS pin to the prescribed value during the reset.

The relationship between the input frequency (fosci) and the SYSCLK, MCLK, and IOCLK multiples and frequencies is shown in Table 3-4-2.

Table 3-4-2 Relationship between the Input Frequency and the SYSCLK, MCLK, and IOCLK Frequencies

Input frequency fosci (MHz)	SYSCLK frequency fsys (MHz)	MCLK frequency fc (MHz)		IOCLK frequency fio (MHz)	
		2x	4x	1x	1/2x
$26.0 \leq \text{fosci} \leq 33.3$	$26.0 \leq \text{fsys} \leq 33.3$	$52.0 \leq \text{fc} \leq 66.6$	—	—	$13.0 \leq \text{fio} \leq 16.6$
$13.0 \leq \text{fosci} \leq 16.6$	$13.0 \leq \text{fsys} \leq 16.6$	—	$52.0 \leq \text{fc} \leq 66.6$	$13.0 \leq \text{fio} \leq 16.6$	—

Chapter 4. Caches

4

4.1   Overview

The MN103002A/MN103002AYB has a 4 Kbytes instruction cache and a 4 Kbytes data cache on chip. Caches are used to overcome the speed difference between external memory and the CPU, increasing the apparent memory access speed. The instruction cache stores line (16-byte) units of instructions requested by the CPU, while the data cache stores line (16-byte) units of data requested by the CPU. If the caches are enabled, instruction fetches in cacheable spaces and data accesses by means of load/store instructions all become eligible for caching.

4.2   Features

The features of the built-in caches of the MN103002A/MN103002AYB are described below.

- Conflicts between instruction accesses and data accesses are avoided since the caches are separated into an instruction cache and a data cache.
- The instruction cache and the data cache are both 4 Kbytes two-way set associative caches.
- Way operation can be selected for the instruction cache and the data cache in any of the following three arrangements:
    (1) 4 Kbytes cache
    (2) 2 Kbytes cache and 2 Kbytes RAM
    (3) 4 Kbytes RAM
- In order to minimize the penalty resulting from cache misses, refilling begins with the word that was missed.
- The data cache can be written either by the write-back method or the write-through method (selectable).
- The entire contents of the cache can be invalidated in one operation.

4.3   Configuration

4.3.1   Instruction Cache

The instruction cache has a 4 Kbytes capacity, and adopts the two-way set associative method for association. The instruction cache consists of two data memory blocks, two tag memory blocks, and an LRU block.

Data memory

The data memory stores instructions in 16-byte units. Each way has 2 Kbytes of data memory, for a total of 4 Kbytes. The data memory line size is 16 bytes, and the number of entries is 128. Instruction transfers from external memory to the instruction cache are handled in 16-byte (128-bit) units, while instruction transfers from the instruction cache to the CPU are handled in 64-bit units. The contents of data memory are not initialized by a reset.

Tag memory

The tag memory has 128 entries, each of which consists of a tag address field (TADD) that stores bits 29 through 11 (19 bits) of the instruction address, a valid bit (V) that indicates whether an entry is valid or not, and a refill bit (R) that indicates whether an instruction transfer (a refill) from external memory to the cache is in progress. In the event of a reset, only the refill bit (R) is cleared. The valid bit (V) is not cleared; it is cleared by the instruction cache invalidate bit in the cache control register (CHCTR).

LRU section

The LRU section stores information that is used to select the entry that should be refilled. The LRU section is shared by way 0 and way 1. The contents of the LRU section can be referenced as part of the tag entries. The LRU section is not initialized by a reset.
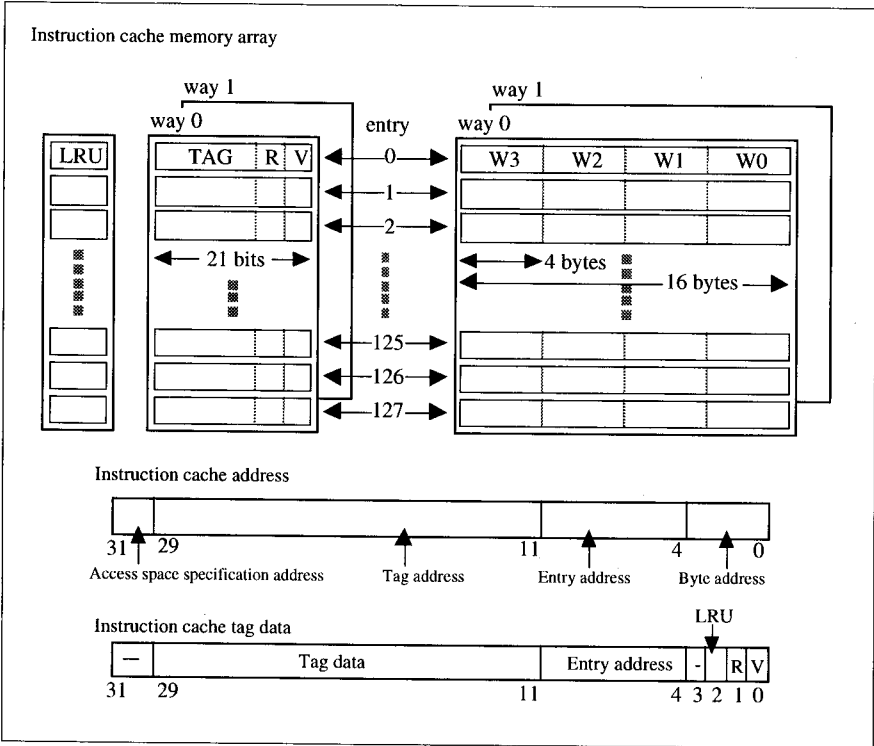
Fig. 4-3-1 Instruction Cache Configuration

4.3.2   Data Cache

The data cache has a 4 Kbytes capacity, and adopts the two-way set associative method for association. The data cache consists of two data memory blocks, two tag memory blocks, and write-back buffers. The data cache can be written either through the write-back method or the write-through method.

Data memory

The data memory stores data in 16-byte units. Each way has 2 Kbytes of data memory, for a total of 4 Kbytes. The data memory line size is 16 bytes, and the number of entries is 128. Data transfers from external memory to the data cache are handled in 16-byte (128-bit) units, while data transfers from the data cache to the CPU are handled in 32-bit units. The contents of data memory are not initialized by a reset.

Tag memory

The tag memory has 128 entries, each of which consists of a tag address field (TADD) that stores bits 30 through 11 (20 bits) of the instruction address, a valid bit (V) that indicates whether an entry is valid or not, a refill bit (R) that indicates whether a data transfer (a refill) from external memory to the data cache is in progress, and a dirty bit (D) that indicates there was a write to the corresponding entry in write-back mode. In the event of a reset, only the refill bit (R) is cleared. The valid bit (V) is not cleared; it is cleared by the data cache invalidate bit in the cache control register (CHCTR).

Write-back buffer

There is a one-line (16-bit) write-back buffer in the data array, and a one-entry write-back buffer in the tag array. The write-back buffers are used to temporarily store write-back data when the data cache is being used in write-back mode.
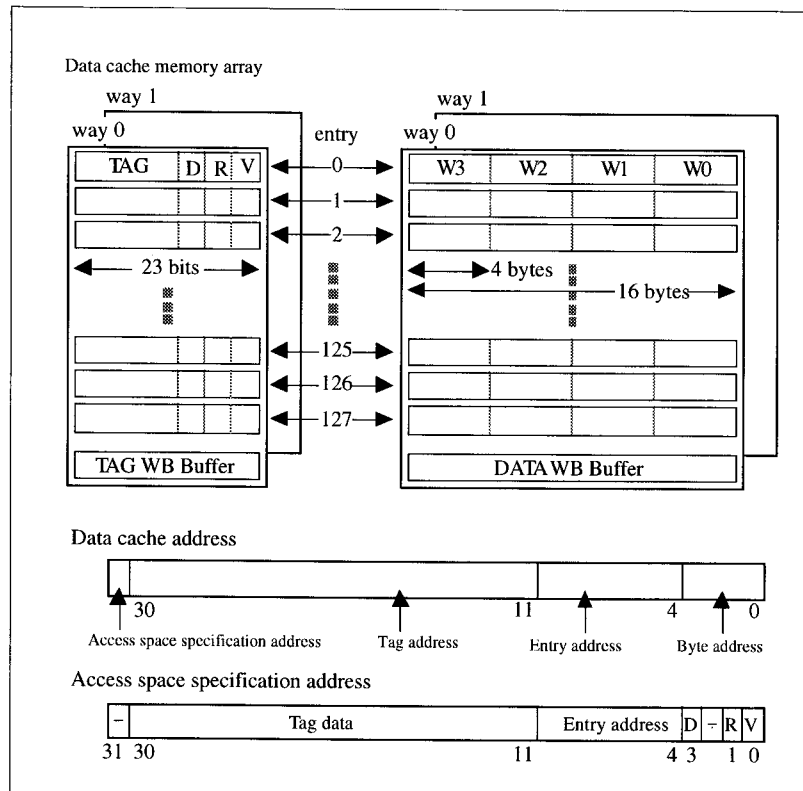
Fig. 4-3-2 Data Cache Configuration

4.4   Description of Registers

The cache control register (CHCTR) stores various settings concerning cache operations (as well as some that do not concern cache operations). In order to use a cache, it is necessary to set the cache control register (CHCTR) and initialize the cache. Writes to the cache control register (CHCTR) are performed with a one-instruction delay. In other words, the cache control register (CHCTR) gains control once the instruction that writes to the cache control register (CHCTR) enters the writing stage in the CPU pipeline. This register is mapped at address x'20000070.

4.4.1   Cache Control Register

Register symbol   CHCTR

Address   x'20000070

Purpose   This register sets the operaton of caches.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	DCW MD1	DCW MD0	-	-	ICW MD1	ICW MD0	INT BR	DC WTMD	DC INV	IC INV	DC BUSY	IC BUSY	DCEN	ICEN
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit No.	Bit name	Description
0	IGEN	Instruction cache enable
		This bit indicates whether or not the instruction cache is to be used.
		0: Instruction cache disabled
		1: Instruction cache enabled
1	DCEN	Data cache enable
		This bit indicates whether or not the data cache is to be used.
		0: Data cache disabled
		1: Data cache enabled
2	ICBUSY	Instruction cache busy
		This bit indicates whether the instruction cache is busy or not. This bit must be checked when accessing the contents of data memory or tag memory directly.
		0: Instruction cache not busy
		1: Instruction cache busy

<Continued>

<Continued>

Bit No.	Bit name	Description
3	DCBUSY	Data cache busy

Data cache busy

This bit indicates whether the data cache is busy or not. This bit must be checked when accessing the contents of data memory or tag memory directly.
0: Data cache not busy
1: Data cache busy

4   ICINV   Instruction cache invalidate

Writing a "1" to this bit invalidates all of the instruction cache ways and entries. This operation is performed by clearing the valid bits (V) for all of the entries in tag memory. This bit always returns a "0" when it is read.

5   DCINV   Data cache invalidate

Writing a "1" to this bit invalidates all of the data cache ways and entries. This operation is performed by clearing the valid bits (V) for all of the entries in tag memory. This bit always returns a "0" when it is read.

6   DCWTMD   Data cache write mode

This bit specifies the data cache write mode.
0: Write-back mode
1: Write-through mode

7   INTBR   Level interrupt base address setting

This bit specifies the interrupt vector base address that is set in the interrupt vector register (IVARn: n = 0 to 6) when a level interrupt is generated.
```
0: x'40000000     1: x'50000000
```

8, 9   ICWMD   Instruction cache way mode

Bit 8 specifies the operation mode for way 0, and bit 9 specifies the operation mode for way 1.
0: Normal operation
1: No refill is performed even if a cache miss occurs

10, 11   reserved

12, 13   DCWMD   Data cache way mode

Bit 12 specifies the operation mode for way 0, and bit 13 specifies the operation mode for way 1.
0: Normal operation
1: No refill is performed even if a cache miss occurs

14, 15   reserved

When invalidating a cache, switching the write mode or performing a way operation, it is necessary to first disable the target cache and then check the busy bit to ensure that the cache is not in operation. That cache operation has halted should also be confirmed when accessing cache data or before performing a purge operation.

4.5 Description of Operation

4.5.1 Instruction Cache

4.5.1.1 Initialization

The instruction cache is disabled when the system is reset. To enable the instruction cache, first set the instruction cache invalidate bit (ICINV) in the cache control register (CHCTR) to invalidate all of the entries, and then set the instruction cache enable bit (ICEN). Examples of an initialization routine are shown below. Note that the cache control register (CHCTR) gains control from the moment that an instruction fetch occurs subsequent to the instruction that writes the cache control register (CHCTR) having entered the writing stage in the CPU pipeline.

When initializing the instruction cache immediately after a reset

```
mov   0x20000070:d,a0
mov   0x00000030:d,d0
mov   d0,(a0)    ; Invalidates the cache (initialization)
mov   0x00000003:d,d0
mov   d0,(a0)    ; Enables the instruction cache/data cache
nop
```

When initializing the instruction cache while it is in operation

```
mov   0x20000070:d,a0
mov   (a0),d0
and   0xfffffffe:d,d0
mov   d0,(a0)    ; Disables the instruction cache
setlb
nop
mov   (a0),d0
btst  0x04,d0    ; Instruction cache busy check
lne
or    0x00000010:d,d0
mov   d0,(a0)    ; Invalidates the instruction cache (initialization)
```

4.5.1.2　Read Operation

Cache Hit Operation

If an instruction is fetched from the cacheable space ($x'40000000$ to $x'7FFFFFFF$) while the instruction cache is enabled, the instruction cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the instruction fetch address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the instruction fetch address and the valid bit (V) of that entry has been set to "1", then a "hit" is said to have occurred in the instruction cache.

If a hit occurs in the instruction cache, the instruction is sent from the corresponding entry (line) in the data memory section to the CPU. Shortage of instructions is prevented with sufficient bandwidth for Bus. i.e. Transfer 64 bits with one access while the process of accessing tag array and instruction fetch requires 2 cycles.

Furthermore, the LRU data is updated simultaneously. If a hit occurs in way 0, the LRU data is set to "1"; if a hit occurs in way 1, the LRU data is set to "0".
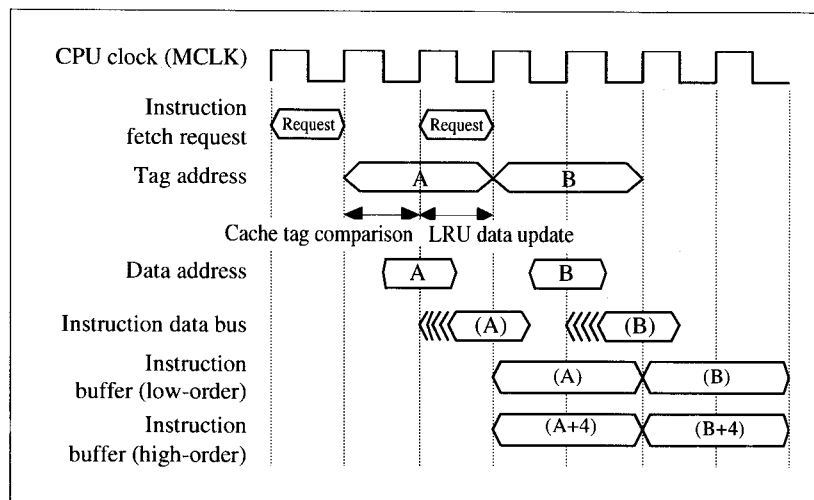
Fig. 4-5-1 Operation When an Instruction Cache Hit Occurs

Cache Miss Operation

If the instruction cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the instruction fetch address as the address, and the value in the tag address field (TADD) of the entry that was accessed does not match the value in the tag field of the instruction fetch address, or if they match but the valid bit (V) of that entry is "0", then an instruction cache "miss" is said to have occurred.

If a miss occurs in the instruction cache, the instruction is fetched from external memory. If an instruction is fetched from external memory, an entry (line) for caching that instruction must be simultaneously allocated in the cache.

First, the refill target way is selected according to the value of the valid bit (V) in the tag array entry that was accessed, the LRU data stored in the LRU field, and the way operation mode that is set in the cache control register (CHCTR). Fig. 4-5-2 shows the flow by which the instruction cache refill target way is selected.
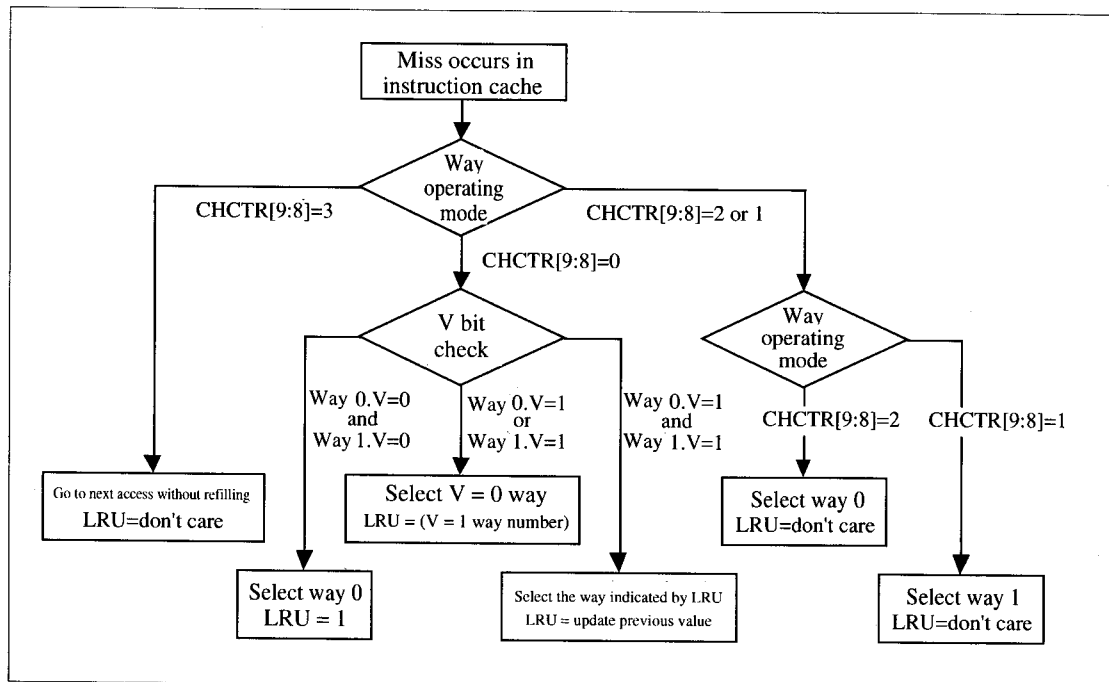
Fig. 4-5-2 Instruction Cache Refill Target Way Selection Flow

Next, after setting the refill bit (R) in the tag entry of the way that was selected, the external bus access (refill) operation, which loads one line of instructions from external memory into cache memory, is initiated. The refill is performed in a burst transfer of four words (16 bytes), which is one line of data, that starts from the access address, length of the word (4 bytes). In the refill sequence, the tag address field (TADD) in the tag array entry is updated, the valid bit (V) is set, and the target line in the data array is updated. In addition, an instruction is passed to the CPU simultaneously. Instructions are passed in 8-byte units; the CPU resumes operations as soon as the upper four bytes of an instruction aligned with an 8-byte boundary are passed.
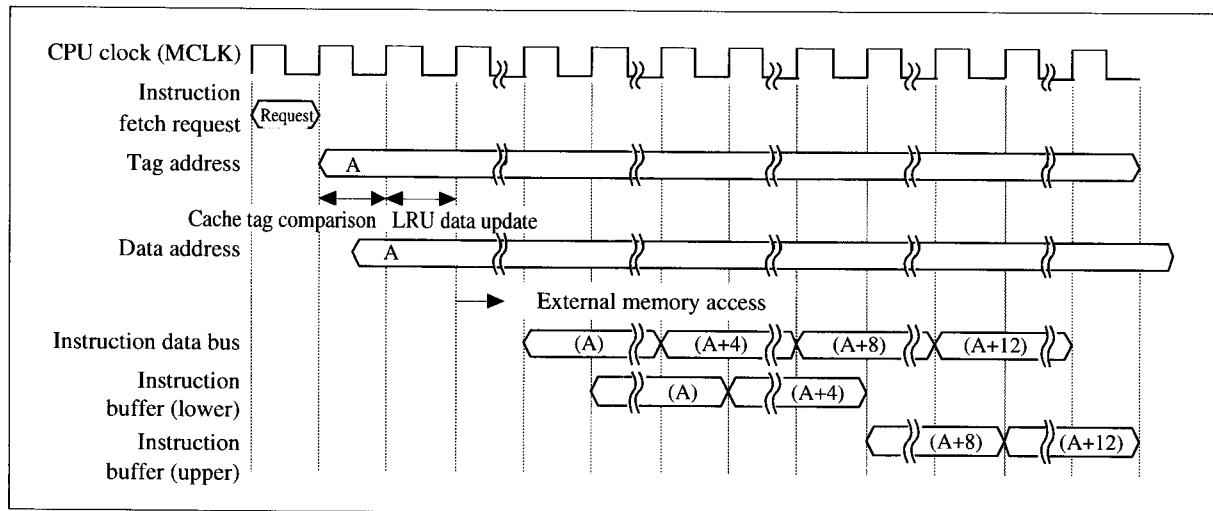
Fig. 4-5-3   Operation When an Instruction Cache Miss Occurs

Disabling of the instruction cache during the instruction cache refill might cause mistake in writing the transmitting word, in the end of refill.

In the case if the uncachable memory area is available

First, finish the instruction disable setting in the uncachable area, also never operate the branch instruction, pointing cachable area, in the following 16 bytes.

In the case if the uncachable memory area is not available

Never operate any branch instruction, pointing cachable area, during the next 16 bytes just after the disable setting of instruction cache. Invalidate the cache entry, containing instruction sets, disabling instruction cache, and a following entry.

4.5.2　Data Cache

4.5.2.1　Initialization

The data cache is disabled when the system is reset. To enable the data cache, first set the data cache invalidate bit (DCINV) in the cache control register (CHCTR) to invalidate all of the entries, and then set the data cache enable bit (DCE). Examples of an initialization routine are shown below. Note that the cache control register (CHCTR) gains control from the moment that a data access occurs subsequent to the instruction that writes the cache control register (CHCTR) having entered the writing stage in the CPU pipeline.

When initializing the instruction cache immediately after a reset
```
mov   0x20000070:d,a0
mov   0x00000030:d,d0
mov   d0,(a0)     ; Invalidates the cache (initialization)
mov   0x20000003:d,d0
mov   d0,(a0)     ; Enables the instruction cache/data cache
nop
```

When initializing the instruction cache while it is in operation
```
mov   0x20000070:d,a0
mov   (a0),d0
and   0xfffffffd:d,d0
mov   d0,(a0)     ; Disables the data cache
setlb
nop
mov   (a0),d0
btst  0x08,d0     ; Data cache busy check
lne
or    0x00000020:d,d0
mov   d0,(a0)     ; Invalidates the data cache (initialization)
```

4.5.2.2 Read Operation

Cache Hit Operation

If data is read from a cacheable space (x'00000000 to x'1FFFFFFF, or x'40000000 to x'7FFFFFFF) while the data cache is enabled, the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address. If the value in the tag address field (TADD) of the entry that was accessed matches the value in the tag field of the data address and the valid bit (V) of that entry has been set to "1", then a "hit" is said to have occurred when reading the data cache.

If a hit occurs when reading the data cache, the data is sent from the corresponding entry (line) in the data memory section to the CPU. Because tag array accesses and data reads are each performed in one cycle in pipeline fashion, then as long as the read accesses keep hitting, consecutive reads can be executed each cycle without any waits.
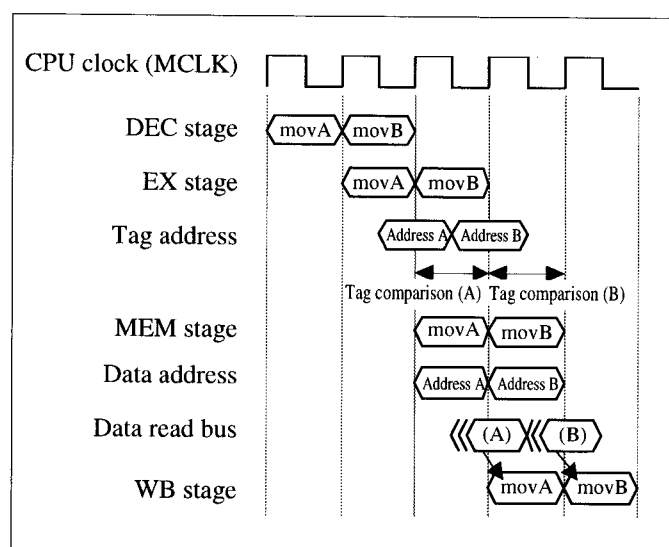
Fig. 4-5-4 Operation When a Data Cache Read Hit Occurs

Cache Miss Operation

If the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the data address, or if they match but the valid bit (V) of that entry is "0", then a data cache "miss" is said to have occurred.

If a miss occurs during a data cache read, the data is accessed in external memory. If data is read from external memory, an entry (line) for caching that data must be simultaneously allocated in the cache. In addition, it is also necessary to check the relationship between data that is expelled from an entry that is to be allocated with the data in external memory in order to maintain the consistency of the data.

First, the refill target way is selected according to the value of the valid bit (V) in the tag array entry that was accessed, the way operation mode that is set in the cache control register (CHCTR), and the information on the way that was selected the last time that data was accessed. Fig. 4-5-5 shows the flow chart how the data cache refill target way is selected.
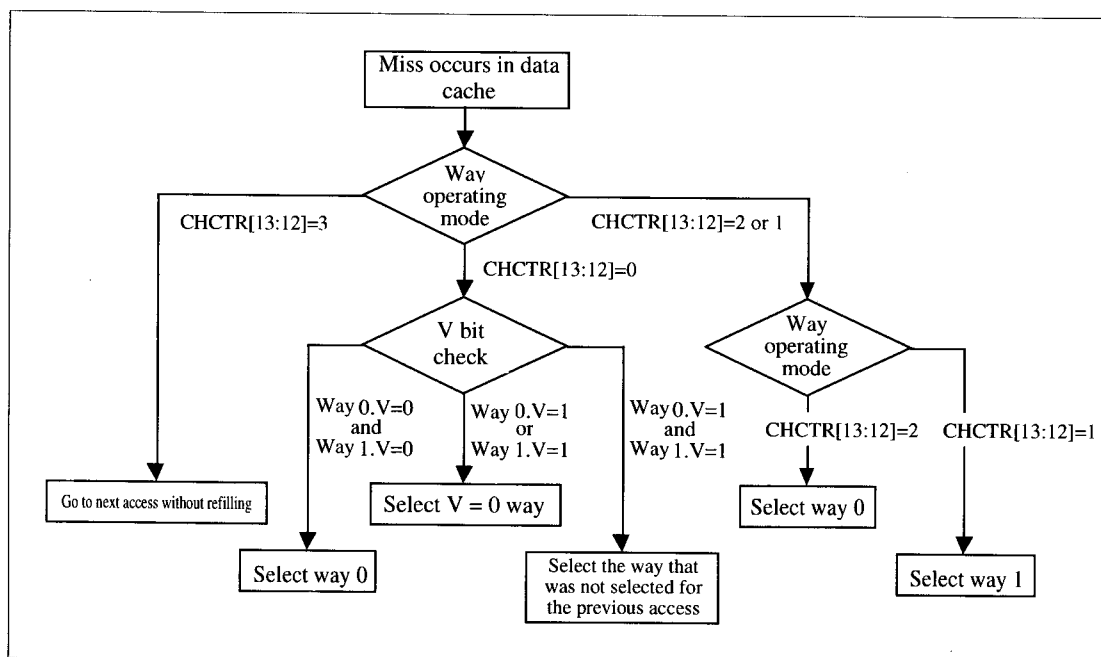
Fig. 4-5-5 Data Cache Refill Target Way Selection Flow

Next, after setting the refill bit (R) in the tag entry of the way that was selected, the values of the dirty bit (D) and the tag address field (TADD) in the same tag entry, and the (one line of) data in the access entry in the data array are saved to the write-back buffer. This save sequence is performed regardless of the data cache writing method that is selected.

Next, the external bus access (refill) operation, which loads one line of data from external memory into cache memory, is initiated. The refill is performed in a burst transfer of four words (16 bytes) that starts from the word (4 bytes) including the access address. In the refill sequence, the tag address field (TADD) in the tag array entry is updated, the valid bit (V) is set, the dirty bit (D) is cleared, and the target line in the data array is updated. In addition, the data is passed to the CPU simultaneously. Data is passed in 4-byte units; the CPU resumes operations as soon as the data is passed.
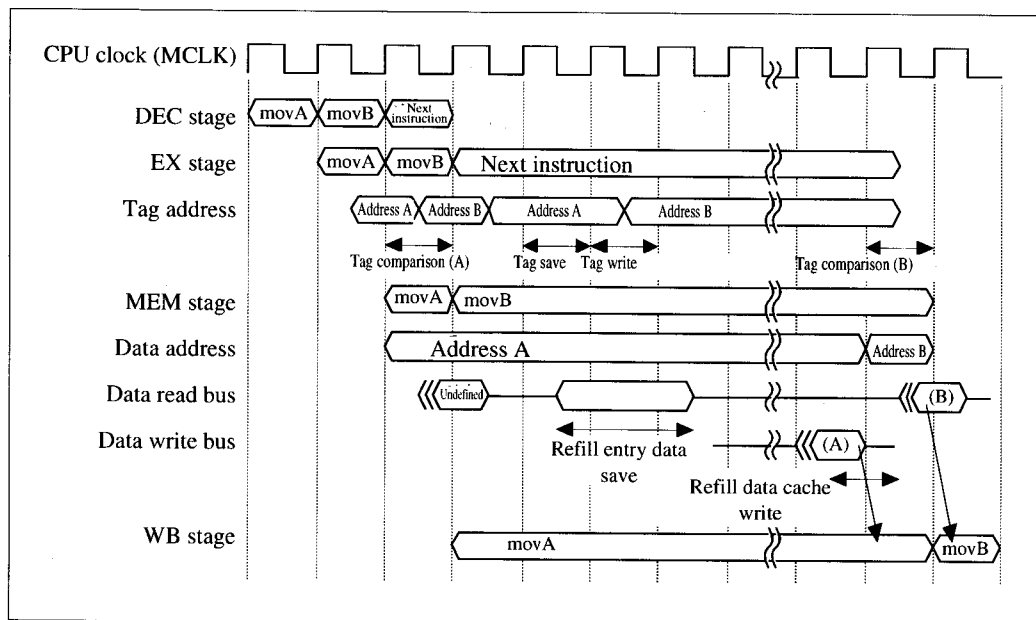
Fig. 4-5-6 Operation When a Data Cache Read Miss Occurs

When the data cache writing method is write-back mode and the dirty bit (D) in the entry that was accessed is set (i.e., is "1"), then after the refill is completed the external bus access for the write-back is initiated and the data that was saved in the write-back buffer is written to external memory. This operation maintains the consistency of the data in the data cache versus the data in the external memory.

4.5.2.3　Write Operation

The data cache supports two types of writing methods: write-back and write-through. The write operation varies, depending on the writing method that is selected.

Write-back Mode

Cache Hit Operation

If data is written to a cacheable space (x' 00000000 to x' 1FFFFFFF, or x' 40000000 to x' 7FFFFFFF) while the data cache is enabled, the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the data address and the valid bit (V) of that entry has been set to "1", then a "hit" is said to have occurred during the write access to the data cache.

If a hit occurs in the write access to the data cache, the data is sent from the CPU to the corresponding entry (line) in the data memory section. At this point, the line in which data was written contains data that is newer than the data that is in the same address in external memory; in this state, the consistency of the data is not being maintained. This state is called the "dirty state," and the dirty bit (D) in the corresponding tag entry is set to "1". This dirty line is written back to external memory when the line is refilled. It is important to remember that, until the line is written back to external memory, the consistency between the data in the data cache and the data in external memory is not being maintained.

Because tag array accesses and data writes are each performed in one cycle in pipeline fashion, then as long as the write accesses keep hitting, consecutive writes can be executed each cycle without any waits.
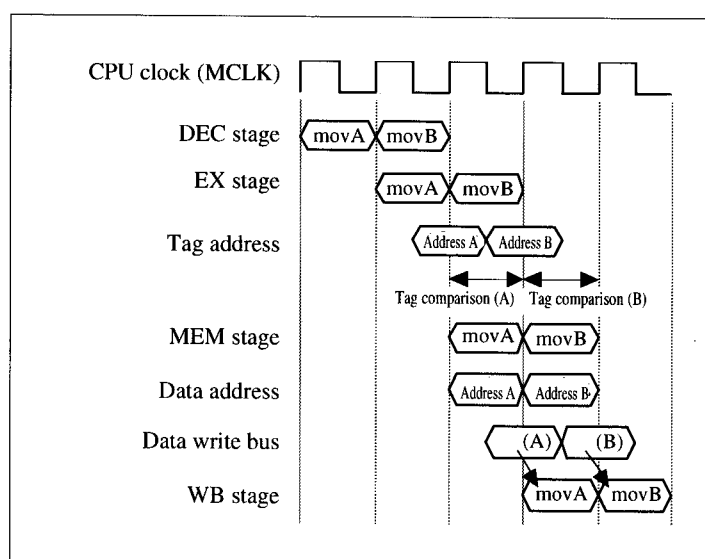
Fig. 4-5-7　Operation When a Data Cache Write Hit Occurs: Write-back Mode

Cache Miss Operation

If the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address, and the value in the tag address field (TADD) of the entry that was accessed does not match the value in the tag field of the data address, or if they match but the valid bit (V) of that entry is "0", then a data cache write access "miss" is said to have occurred.

In write-back mode, even if the data cache write access results in a miss, the data is written in the cache. Therefore, it is necessary to allocate an entry (line) in the cache for writing the data. In addition, it is also necessary to check the relationship between data that is expelled from an entry that is to be allocated with the data in external memory in order to maintain the consistency of the data.

First, the refill target way is selected according to the value of the valid bit (V) in the tag array entry that was accessed, the way operation mode that is set in the cache control register (CHCTR), and the information on the way that was selected the last time that data was accessed. The flow by which the refill target way is selected is the same as when a data cache read miss occurs. (Refer to Fig. 4-5-5.)

Next, after setting the refill bit (R) in the tag entry of the way that was selected, the values of the dirty bit (D) and the tag address field (TADD) in the same tag entry, and the (one line of) data in the access entry in the data array are saved to the write-back buffer. This save sequence is performed regardless of the data cache writing method that is selected.

Next, the external bus access (refill) operation, which loads one line of data from external memory into cache memory, is initiated. The refill is performed in a burst transfer of four words (16 bytes) that starts from the word (4 bytes) that includes the access address. In the refill sequence, the tag address field (TADD) in the tag array entry is updated, the valid bit (V) and the dirty bit (D) are set, and the target line in the data array is updated.
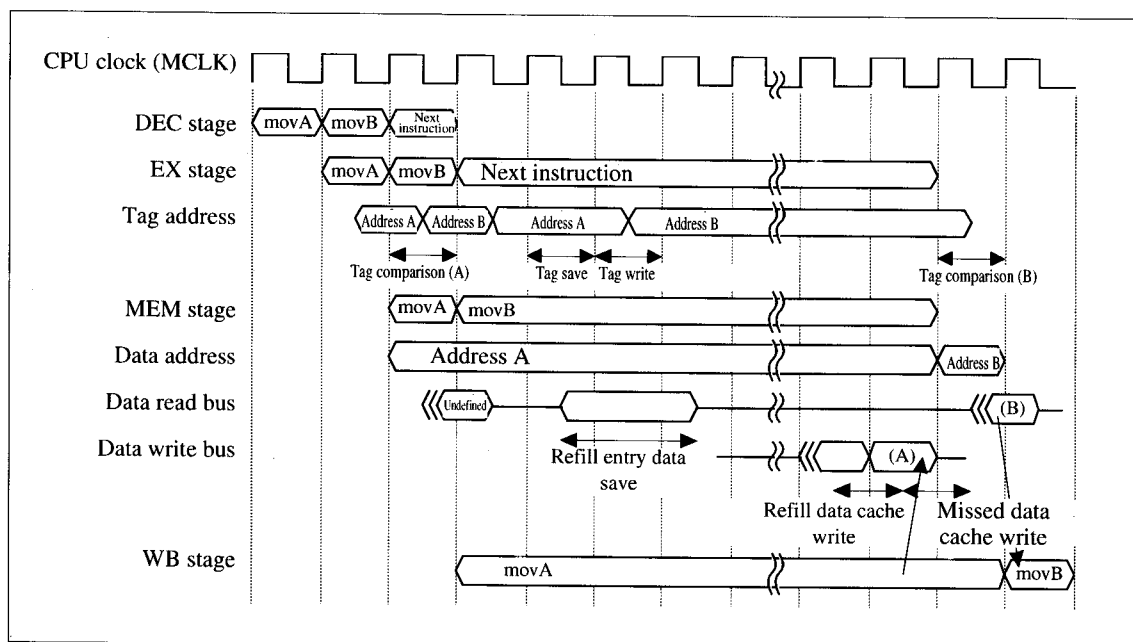
Fig. 4-5-8 Operation When a Data Cache Write Miss Occurs: Write-back Mode

If the dirty bit (D) in the entry that was accessed is set (i.e., "1"), then after the refill is completed the external bus access for the write-back is initiated and the data that was saved in the write-back buffer is written to external memory. This operation maintains the consistency of the data in the data cache versus the data in the external memory.

In the data cache way operation mode settings, if both ways are set to make no refill even if a cache miss occurs, the write operation is not performed and the CPU executes the next instruction.

Write-through Mode

When using the data cache in write-through mode, use one of the methods shown below.

[1] Using the data cache as 4 KB cache system,

Case1: If the stack is located whithin the uncachable space,

(a) Before any access to the IO space(x'20000000 to x'3FFFFFFF), execute the instruction that reads from the uncachable space(x'80000000 to x'BFFFFFFF).

Case2: If the stack is located within the cachable space,

(a) Before any access to the IO space(x'20000000 to x'3FFFFFFF), execute the instruction that reads from an uncachable space(x'80000000 to x'BFFFFFFF).

(b) Just before rti instructions (return from interrupt handler), execute dummy accesses shown below.

```
mov  (0x0:b, SP), d0 ;
mov  (0x4:b, SP), d0 ;
mov  (a0), d0        ;    Make sure that address "a0" is within
                          the uncachable space.
```

(c) It is necessary that the instruction that reads from the uncachable space is placed at the return address of the subroutine call instruction (call).

[2] If either of two ways is need to be lockd, lock the way 1.

Cache Hit Operation

If data is written to a cacheable space (x'00000000 to x'1FFFFFFF, or x'40000000 to x'7FFFFFFF) while the data cache is enabled, the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address. If the value in the tag address field (TADD) of the accessed entry matches the value in the tag field of the data address and the valid bit (V) of that entry has been set to "1", then a "hit" is said to have occurred during the write access to the data cache.

If a hit occurs in the write access to the data cache, the data is sent from the CPU to the corresponding entry (line) in the data memory section. At the same time, data is written to the corresponding address in external memory. Because the data is written simultaneously to the data cache and to the external memory, consistency between the data in the data cache and the data in the external memory is maintained.

Operations during write accesses to the data cache in write-through mode are closely related data cache way mode setting. The write to the data cache is performed regardless of which operating mode is selected. With regard to the write to external memory, however, the case where one way was set to the mode in which a refill is not made even if a cache miss occurs is an exception, and the bug of writing to external memory for a write to the way that was set to the mode in which a refill is not made even if a cache miss occurs has not been incorporated into the specifications. All other writes entail writes to external memory.

In write-through mode, the CPU halts operations until the writing of data to external memory has been completed.
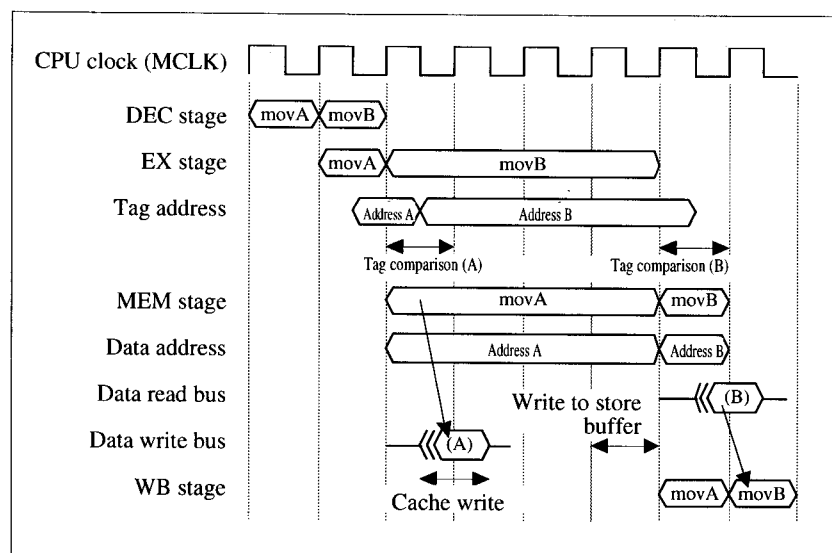
Fig. 4-5-9 Operation When a Data Cache Write Hit Occurs: Write-through Mode

Cache Miss Operation

If the data cache tag array is accessed, using the tag entry address field (bits 10 to 4) of the data address as the address, and the value in the tag address field (TADD) of the accessed entry does not match the value in the tag field of the data address, or if they match but the valid bit (V) of that entry is "0", then a data cache write access "miss" is said to have occurred.

In write-through mode, if the data cache write access results in a miss, the data is written to external memory, and is not written in the data cache. In addition, no refill operation is performed.
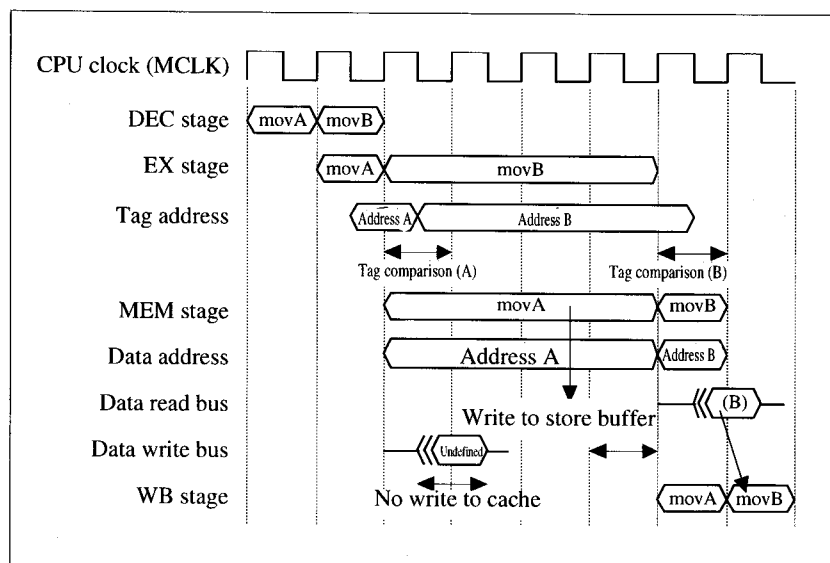
Fig. 4-5-10 Operation When a Data Cache Write Miss Occurs: Write-through Mode

> **!** When switching between write-back mode and write-through mode while in operation, be certain to first invalidate the entire contents of the data cache through the cache control register.

4.5.2.4 Consistency between the Cache and the External Memory

In order to maintain consistency between the data stored in the data cache and the data in the external memory, it is necessary to write the data that is stored in the data cache back to the external memory. This operation is called "purging," and in the MN103002A/MN103002AYB is performed by accessing specific addresses. (Consistency must be guaranteed by software.)

As shown in the memory map in Fig. 4-5-11, purge addresses are assigned for each entry in the data cache. Any individual entry can be written back to external memory by making a data access (read or write) to the corresponding address. For example, in order to purge entry 0 of way 0, all that is necessary is to perform a data access on address x'28400000.

```
mov         (0x28400000),d0
or
mov         d0,(0x28400000)
```

The most efficient way to purge multiple entries is to use the setlb instruction, etc., to form a loop.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x'2840000X													Way 0, entry 0			
x'2840001X													Way 0, entry 0			
x'284007EX													Way 0, entry 126			
x'284007FX													Way 0, entry 127			
x'2840100X													Way 1, entry 0			
x'2840101X													Way 1, entry 0			
x'284017EX													Way 1, entry 126			
x'284017FX													Way 1, entry 127			

Fig. 4-5-11 Purge Addresses

Before performing a purge, disable the data cache and then confirm that the data cache busy bit (DCBUSY) in the cache control register (CHCTR) has been reset (to "0").

The data accesses that are made to the specific addresses that are used to initiate purges are dummy accesses. If the data access is a read access, the data that is returned is undefined. If the data access is a write access, no data is actually written anywhere.

4.5.3   Way Operation Mode

The operation mode can be set for each way for both the instruction cache and the data cache through the cache control register (CHCTR). In normal operation mode, the way functions as a cache, but in the mode in which the refill operation is not performed even if a cache miss occurs, each way can be used as RAM. The RAM in this case is positioned as a cache in which a hit is always generated. Therefore, just as with a normal cache, the access to the contents of RAM that accompanies a tag access requires that the tag access generates a hit. (In other words, when using a way as RAM, it is necessary to manipulate the contents of the tag so that the necessary address generates a hit.)

A way that is set in the mode that does not perform a refill even in the event of a cache miss retains a variety of statuses from the moment that the mode was set (This is equivalent to locking the cache). Even if a cache miss occurs, the way is not selected as the target of the refill operation. Regarding a way that is used as RAM, the contents of the cache cannot be changed by a refill, but after disabling the cache, an access to the I/O space can be used to change the data in the tag array and the data in the data array.

If all ways are set in the mode that does not perform a refill even in the event of a cache miss, set tag data that will definitely result in a cache hit when the cacheable space is accessed. With these settings, the read data can not be guaranteed in the event of an access to a cacheable space that results in a cache miss (The CPU does not stop operating).

4.5.4   External Bus Access During Cache Operations

Although bus accesses outside of the MN103002A/MN103002AYB can be made freely during cache operations, if a cache miss occurs, external bus access is not possible until the refilling of the cache from external memory and the transfer of data from the cache to external memory due to a write-back/purge operation is completed.

Resources mapped in the I/O space (x'20000000 to x'3FFFFFFF) within the MN103002A/MN103002AYB can be accessed while a refill, write-back, or purge operation is in progress. (However, this does not include accesses to cache entries during cache operations.)

4.6　Cache Entry Address Assignments

The cache memory tag array and data array are mapped in the internal I/O area, and can be read and written directly. The access size of these arrays is the word (32 bits). It is essential to note that if the contents of the tag array and the data array are overwritten while the area is being used as a cache, the contents of external memory will no longer match the contents of the cache.

4.6.1　Tag Array

Instruction cache tag array I/O memory

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x'2810000X													Way 0, entry 0			
x'2810001X													Way 0, entry 1			
x'281007EX													Way 0, entry 126			
x'281007FX													Way 0, entry 127			
x'2810100X													Way 1, entry 0			
x'2810101X													Way 1, entry 1			
x'281017EX													Way 1, entry 126			
x'281017FX													Way 1, entry 127			

Data cache tag array I/O memory map

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x'2830000X													Way 0, entry 0			
x'2830001X													Way 0, entry 1			
x'283007EX													Way 0, entry 126			
x'283007FX													Way 0, entry 127			
x'2830100X													Way 1, entry 0			
x'2830101X													Way 1, entry 1			
x'283017EX													Way 1, entry 126			
x'283017FX													Way 1, entry 127			

Fig. 4-6-1　Tag Array Address Assignments

4.6.2 Data Array

Instruction cache data array I/O memory map

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x'2800000X	Way 0, entry 0,offset 3				Way 0, entry 0,offset 2				Way 0, entry 0,offset 1				Way 0, entry 0,offset 0			
x'2800001X	Way 0, entry 1,offset 3				Way 0, entry 1,offset 2				Way 0, entry 1,offset 1				Way 0, entry 1,offset 0			
x'280007EX	Way 0, entry 126,offset 3				Way 0, entry 126,offset 2				Way 0, entry 126,offset 1				Way 0, entry 126,offset 0			
x'280007FX	Way 0, entry 127,offset 3				Way 0, entry 127,offset 2				Way 0, entry 127,offset 1				Way 0, entry 127,offset 0			
x'2800100X	Way 1, entry 0,offset 3				Way 1, entry 0,offset 2				Way 1, entry 0,offset 1				Way 1, entry 0,offset 0			
x'2800101X	Way 1, entry 1,offset 3				Way 1, entry 1,offset 2				Way 1, entry 1,offset 1				Way 1, entry 1,offset 0			
x'280017EX	Way 1, entry 126,offset 3				Way 1, entry 126,offset 2				Way 1, entry 126,offset 1				Way 1, entry 126,offset 0			
x'280017FX	Way 1, entry 127,offset 3				Way 1, entry 127,offset 2				Way 1, entry 127,offset 1				Way 1, entry 127,offset 0			

Data cache data array I/O memory map

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x'2820000X	Way 0, entry 0,offset 3				Way 0, entry 0,offset 2				Way 0, entry 0,offset 1				Way 0, entry 0,offset 0			
x'2820001X	Way 0, entry 1,offset 3				Way 0, entry 1,offset 2				Way 0, entry 1,offset 1				Way 0, entry 1,offset 0			
x'282007EX	Way 0, entry 126,offset 3				Way 0, entry 126,offset 2				Way 0, entry 126,offset 1				Way 0, entry 126,offset 0			
x'282007FX	Way 0, entry 127,offset 3				Way 0, entry 127,offset 2				Way 0, entry 127,offset 1				Way 0, entry 127,offset 0			
x'2820100X	Way 1, entry 0,offset 3				Way 1, entry 0,offset 2				Way 1, entry 0,offset 1				Way 1, entry 0,offset 0			
x'2820101X	Way 1, entry 1,offset 3				Way 1, entry 1,offset 2				Way 1, entry 1,offset 1				Way 1, entry 1,offset 0			
x'282017EX	Way 1, entry 126,offset 3				Way 1, entry 126,offset 2				Way 1, entry 126,offset 1				Way 1, entry 126,offset 0			
x'282017FX	Way 1, entry 127,offset 3				Way 1, entry 127,offset 2				Way 1, entry 127,offset 1				Way 1, entry 127,offset 0			

Fig. 4-6-2 Data Array Address Assignments

Chapter 5.   Bus Controller

5

5.1   Overview

In this LSI, the bus controller (BC) controls interfacing between the CPU core, on-chip I/O (peripherals), and devices external to the chip. The bus controller also has DMA control functions and handles arbitration between the internal and external buses. The bus controller outputs eight chip select signals, RAS/CAS signals, and other signals for an external bus interface, permitting ROM, SRAM, DRAM, and other peripheral LSIs to be connected directly to this LSI without the need for any additional external circuits.

5.2   Features

The features of the bus controller are described below.

● High-speed control of the internal and external buses through the CPU clock (MCLK) is possible.
- Synchronous mode and asynchronous mode (synchronized with MCLK) are supported for the internal I/O bus and external buses. In asynchronous mode, wait control in units of CPU cycles is possible.

● External memory is managed through partitioning into 8 blocks.
- There is a chip select output for each block.
- The bus width can be set to 16 or 32 bits for each block.
- Blocks 2 to 7 permit switching between fixed wait state insertion and handshaking.
- Blocks 0 to 5 permit switching between synchronous mode and asynchronous mode.
- Blocks 1 to 4 can be used as DRAM space.

● DRAM interface
- Address multiplexing function. (The low address shift amount can be selected from between 8 bits to 11 bits.)
- Supports two byte specification methods. (CAS or WE)
- Support for high-speed page mode. (Supports the page mode mix cycle DRAM.)
- Support for CAS-before-RAS refresh. (Programmable refresh cycle)

● Avoids time penalty during storage operations through use of store buffer (one word).
- Support for storage in on-chip peripheral circuits and external devices
- When the storage buffer is empty, storage operations are completed with no wait states, and the CPU can execute subsequent processing.

5.3   Bus Configuration

Fig. 5-3-1 shows the bus configuration in the MN103002A/MN103002AYB. The chip's internal buses are the ICH bus between the CPU core and the instruction cache, the DCH bus between the CPU core and the data cache, the BC bus between the CPU core and the bus controller, and the I/O bus between the bus controller and on-chip I/O. The EX bus is an external bus (user bus). Table 5-3-1 lists the characteristics of each bus.
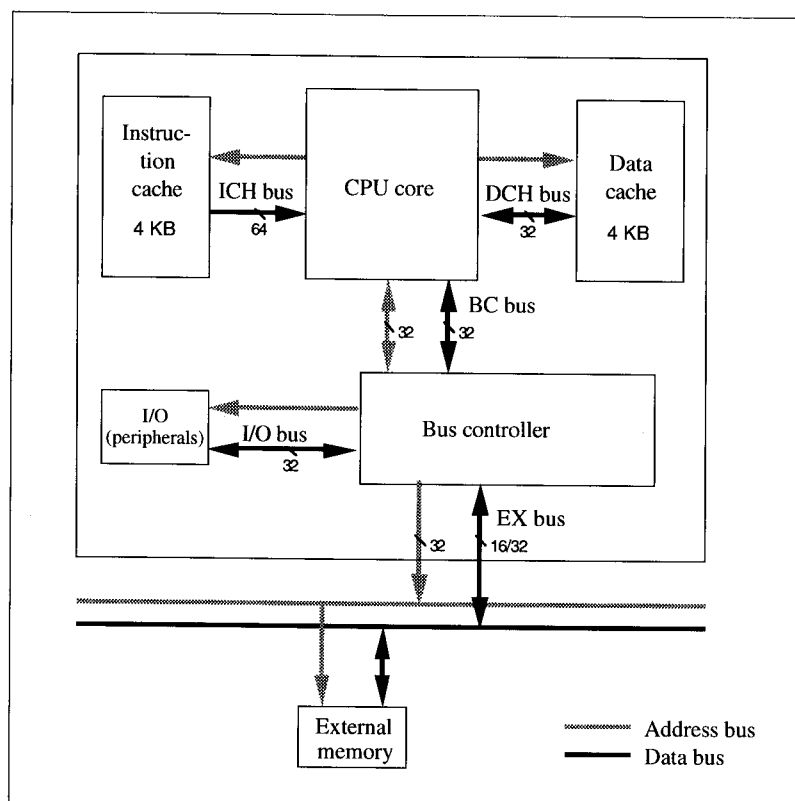
Fig. 5-3-1 Bus Configuration Diagram

Table 5-3-1 Characteristics of Each Bus

Bus name	Blocks	Bus width	Operating clock	
ICH bus	CPU - instruction cache	64	MCLK [*1]	
DCH bus	CPU - data cache	32	MCLK [*1]	
BC bus	CPU — BC	32	MCLK [*1]	
I/O bus	BC — internal I/O	32	IOCLK [*2]	[synchronous mode]
			MCLK [*1]	[asynchronous mode]
EX bus (external bus)	BC — external memory	16/32	SYSCLK [*3]	[synchronous mode]
			MCLK [*1]	[asynchronous mode]

[*1]: The MCLK frequency is either four times the input frequency (when FRQS = high) or two times the input frequency (when FRQS = low).

[*2]: IOCLK is the operation clock for the internal I/O block. The IOCLK frequency is either the same as the input frequency (when FRQS = high) or one-half the input frequency (when FRQS = low).

[*3]: The SYSCLK frequency is the same as the input frequency.

5.4 Block Diagram

Fig. 5-4-1 shows the block diagram for the bus controller. The bus controller consists of a controller, a CPU interface, an on-chip I/O interface, an external device interface, and a DMA controller.
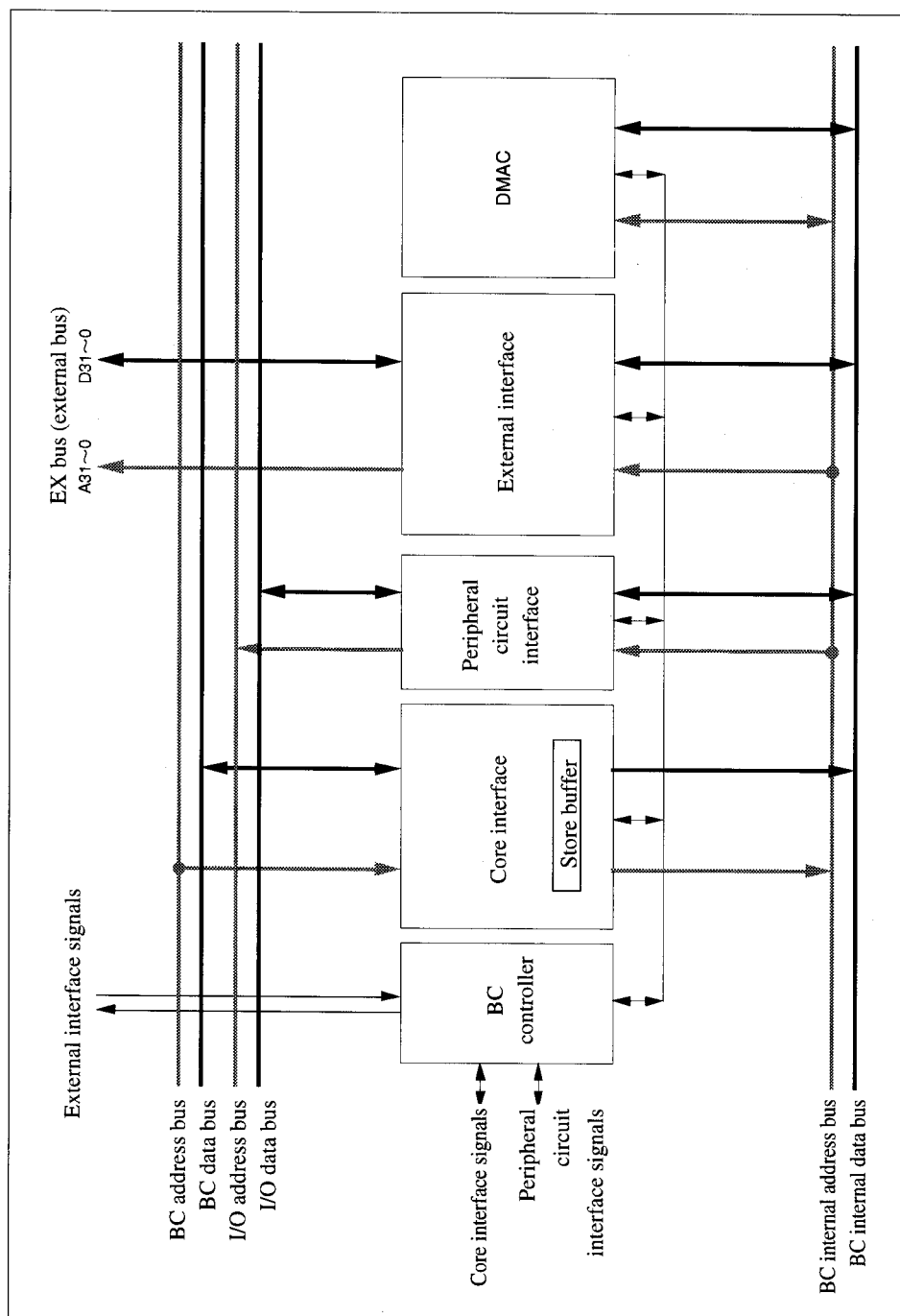
Fig. 5-4-1   Bus Controller Block Diagram

5.5 Pin Functions

Table 5-5-1 External Pin Functions Relating to the Bus Controller

Pin name	Input/ output	Number of pins	Function
OSCI	Input	1	Oscillator input pin (13.0 MHz to 33.3 MHz)
OSCO	Output	1	Oscillator output pin (13.0 MHz to 33.3 MHz)
FRQS	Input	1	Input frequency setting (1: 13.0 MHz to 16.6 MHz; 0: 26.0 MHz to 33.3 MHz)
SYSCLK	Output	1	System clock output (same frequency as the input clock)
A31-0	Output	32	Memory address output (Row/column address multiplexed output when DRAM is connected)
D31~0	Input/ output	32	Memory data input/output (D31 to 16 for a 16-bit bus)
RAS4~1	Output	4	DRAM RAS signal
CAS3~0	Output	4	DRAM CAS signals [*1]
CS7~0	Output	8	Chip select signals
RE	Output	1	Memory read signal
WE3~0	Output	4	Memory write signals (output in byte units) [*2]
DK	Input	1	Data acknowledge signal
RD/WT	Output	1	Read/write status signal
SIZE1~0	Output	2	Access size information notification signal (00: 1 byte; 01: 2 bytes; 10: 3 bytes; 11: 4 bytes)
BR	Input	1	Bus authority request signal
BG	Output	1	Bus authority release signal
BMODE	Input	1	Block 0 data bus width specification (0: 16 bits; 1: 32 bits)
DMR3~0	Input	4	DMA request signals
DMK3~0	Output	4	DMA acknowledge signals
MMODE	Output	1	Test pin

[*1]: CAS3 corresponds to D31 to 24, CAS2 corresponds to D23 to 16, CAS1 corresponds to D15 to 8, and CAS0 corresponds to D7 to 0.

[*2]: WE3 corresponds to D31 to 24, WE2 corresponds to D23 to 16, WE1 corresponds to D15 to 8, and WE0 corresponds to D7 to 0.

Note: CS3 to 1 and CAS3 to 1 are shared pins, and CS7 and A31, and CS6 to 4 and A28 to 26 are shared pins.

5.6 Description of Registers

Table 5-6-1 lists the bus controller registers. The settings of these registers are used in wait state control, DRAM interface control, etc.

Table 5-6-1 List of BC Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'32000010	I/O bus control register	IOBCTR	16	x'0F70	8, 16
x'32000020	Memory control register 0	MEMCTR0	16	x'XXXX *1	8, 16
x'32000022	Memory control register 1	MEMCTR1	16	x'1F20	8, 16
x'32000024	Memory control register 2	MEMCTR2	16	x'1F20	8, 16
x'32000026	Memory control register 3	MEMCTR3	16	x'1F20	8, 16
x'32000028	Memory control register 4	MEMCTR4	16	x'1F20	8, 16
x'3200002A	Memory control register 5	MEMCTR5	16	x'0020	8, 16
x'3200002C	Memory control register 6	MEMCTR6	16	x'0020	8, 16
x'3200002E	Memory control register 7	MEMCTR7	16	x'0020	8, 16
x'32000040	DRAM control register	DRAMCTR	16	x'0000	8, 16
x'32000042	Refresh count register	REFCNT	16	x'FFFF	8, 16

*1: Depends on the value of FRQS.

5.6.1 I/O Bus Control Register

Register symbol: IOBCTR

Address: x' 32000010

Purpose: Sets the I/O bus mode and the number of wait states.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	IOP WC3	IOP WC2	IOP WC1	IOP WC0	-	IO WC2	IO WC1	IO WC0	-	-	-	IO BM
When reset	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IOBM	I/O bus mode 0: Synchronous mode (synchronized with IOCLK) 1: Asynchronous mode (synchronized with MCLK)
4	IOWC0	Number of I/O bus wait state insertions (on-chip I/O other than the I/O port section) (LSB)
5	IOWC1	Number of I/O bus wait state insertions (on-chip I/O other than the I/O port section)
6	IOWC2	Number of I/O bus wait state insertions (on-chip I/O other than the I/O port section) (MSB) The relationship between the value of IOWC2 to 0 and the number of wait states is shown below.

Synchronous mode [IOBM = 0] (number of wait states counted by IOCLK)	Asynchronous mode [IOBM = 1] (number of wait states counted by MCLK)

(Ignore the lowest two bits)
 	
    0xx :   0 wait states
    1xx :   1 wait states
	000 : Setting prohibited 001 : Setting prohibited
	010 : 2 wait states 011 : 3 wait states
	100 : 4 wait states 101 : 5 wait states
	110 : 6 wait states 111 : 7 wait states

<Continued>

<Continued>

Bit No.	Bit name	Description
8	IOPWC0	Number of I/O bus wait state insertions (I/O port section) (LSB)
9	IOPWC1	Number of I/O bus wait state insertions (I/O port section)
10	IOPWC2	Number of I/O bus wait state insertions (I/O port section)
11	IOPWC3	Number of I/O bus wait state insertions (I/O port section) (MSB)

The relationship between the value of IOPWC3 to 0 and the number of wait states is shown below.

Synchronous mode [IOBM = 0] (number of wait states counted by IOCLK)	Asynchronous mode [IOBM = 1] (number of wait states counted by MCLK)
(Ignore the lowest two bits) 00xx : 0 wait states 01xx : 1 wait states 10xx : 2 wait states 11xx : 3 wait states	Not supported

Note that accesses to the I/O port section can only be made in synchronous mode.

Accesses to on-chip I/O other than the I/O port section are made in synchronous mode with 1 wait state after reset mode is released. Accesses to the I/O port section are made in synchronous mode with 3 wait states.

5.6.2　Memory Control Registers

5.6.2.1　Memory Control Register 0

Register symbol:　　MEMCTR0

Address:　　　　　x' 32000020

Purpose:　　　　　Sets the bus mode and the number of wait states to be inserted for external memory
space block 0.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	FRQ	-	-	B0 WC4	B0 WC3	B0 WC2	B0 WC1	B0 WC0	-	-	-	-	-	B0 BM	-	-
When reset	0/1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
Access	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R

Bit No.	Bit name	Description
2	B0BM	Block 0 bus mode
		0: Synchronous mode (synchronized with SYSCLK)
		1: Asynchronous mode (synchronized with MCLK)
8	B0WC0	Number of block 0 wait state insertions (LSB)
9	B0WC1	Number of block 0 wait state insertions
10	B0WC2	Number of block 0 wait state insertions
11	B0WC3	Number of block 0 wait state insertions
12	B0WC4	Number of block 0 wait state insertions (MSB)

The relationship between the value of B0WC4 to 0 and the number of wait states is shown below.

(1) When not using the in-circuit emulator (PX-ICE103002)

Basic bus cycle	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQ = 0	x0000 : Setting prohibited x0001 : 1 wait states x0010 : 2 wait states x0011 : 3 wait states : x1111 : 15 wait states	00000 : Setting prohibited 00001 : 1 wait states 00010 : 2 wait states 00011 : 3 wait states : 11111 : 31 wait states
FRQ = 1		00000 : Setting prohibited 00001 : Setting prohibited 00010 : Setting prohibited 00011 : 3 wait states : 11111 : 31 wait states

<Continued>

<Continued>

(2) When using the in-circuit emulator (PX-ICE103002)

[1] When using the emulation function

External input pin	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQS = L	x0000 : Setting prohibited x0001 : Setting prohibited x0010 : 2 wait states x0011 : 3 wait states : x1111 : 15 wait states	00000 : Setting prohibited : 00011 : Setting prohibited 00100 : 4 wait states : 11111 : 31 wait states
FRQS = H	x0000 : Setting prohibited x0001 : 1 wait states x0010 : 2 wait states x0011 : 3 wait states : x1111 : 15 wait states	

[2] When using the external trace function

External input pin	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQS = L	x0000 : Setting prohibited x0001 : 1 wait states x0010 : 2 wait states	00000 : Setting prohibited 00001 : Setting prohibited 00010 : Setting prohibited
FRQS = H	x0011 : 3 wait states : x1111 : 15 wait states	00011 : 3 wait states : 11111 : 31 wait states

Bit No.	Bit name	Description
15	FRQ	Bus sequence setting 0: Basic 2 cycles 1: Basic 4 cycles For details, refer to the individual timing charts.

The initial value of FRQ is determined by FRQS.

After reset mode is released, block 0 is set to synchronous mode with 15 wait states, and the bus width is as specified by the BMODE pin. Block 0 cannot be used as a DRAM space.

5.6.2.2 Memory Control Register 1

Register symbol: MEMCTR1

Address: x' 32000022

Purpose: Sets the bus mode and the number of wait states to be inserted for external memory space block 1.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	B1 WC4	B1 WC3	B1 WC2	B1 WC1	B1 WC0	-	B1 CAS	B1 BW	-	-	B1 BM	-	B1 DRAM
When reset	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0
Access	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R	R/W

Bit No.	Bit name	Description
0	B1DRAM	Block 1 DRAM space setting 0: Do not use this block as a DRAM space 1: Use this block as a DRAM space
2	B1BM	Block 1 bus mode 0: Synchronous mode (synchronized with SYSCLK) 1: Asynchronous mode (synchronized with MCLK)
5	B1BW	Block 1 bus width 0: 16 bits 1: 32 bits
6	B1CAS	Block 1 DRAM byte specification 0: Specified by WE3 to 0 1: Specified by CAS3 to 0
8	B1WC0	Number of block 1 wait state insertions (LSB)
9	B1WC1	Number of block 1 wait state insertions
10	B1WC2	Number of block 1 wait state insertions
11	B1WC3	Number of block 1 wait state insertions
12	B1WC4	Number of block 1 wait state insertions (MSB)

After reset mode is released, block 1 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. When using block 1 as a DRAM space, the number of wait states is as indicated by the setting of the WC bit in the DRAM control register. However, even when using block 1 as a DRAM space, be sure not to set a prohibited value in B1WC4 to 0.

DRAM mode is not valid when the DRAME bit in the DRAMCTR register is "0" and BIDRAM is "1".

> The number of wait states that are inserted is the same as set in bits B0WC4 to 0. Refer to the description of memory control register 0 (MEMCTR0).

5.6.2.3 Memory Control Register 2

Register symbol: MEMCTR2

Address: x' 32000024

Purpose: Sets the bus mode and the number of wait states to be inserted for external memory space block 2.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	B2 WC4	B2 WC3	B2 WC2	B2 WC1	B2 WC0	-	B2 CAS	B2 BW	-	-	B2 BM	B2 WM	B2 DRAM
When reset	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0
Access	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	B2DRAM	Block 2 DRAM space setting 0: Do not use this block as a DRAM space 1: Use this block as a DRAM space
1	B2WM	Block 2 wait mode 0: Fixed wait state insertion 1: Handshaking
2	B2BM	Block 2 bus mode 0: Synchronous mode (synchronized with SYSCLK) 1: Asynchronous mode (synchronized with MCLK)
5	B2BW	Block 2 bus width 0: 16 bits 1: 32 bits
6	B2CAS	Block 2 DRAM byte specification 0: Specified by WE3 to 0 1: Specified by CAS3 to 0
8	B2WC0	Number of block 2 wait state insertions (LSB)
9	B2WC1	Number of block 2 wait state insertions
10	B2WC2	Number of block 2 wait state insertions
11	B2WC3	Number of block 2 wait state insertions
12	B2WC4	Number of block 2 wait state insertions (MSB)

After reset mode is released, block 2 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. When using block 2 as a DRAM space, the number of wait states is as indicated by the setting of the WC bit in the DRAM control register. However, even when using block 2 as a DRAM space, be sure not to set a prohibited value in B2WC4 to 0.

> **!** The number of wait states that are inserted is the same as set in bits B0WC4 to 0. Refer to the description of memory control register 0 (MEMCTR0).

5.6.2.4   Memory Control Register 3

Register symbol: MEMCTR3

Address: x' 32000026

Purpose: Sets the bus mode and the number of wait states to be inserted for external memory space block 3.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	B3 WC4	B3 WC3	B3 WC2	B3 WC1	B3 WC0	-	B3 CAS	B3 BW	-	-	B3 BM	B3 WM	B3 DRAM
When reset	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0
Access	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	B3DRAM	Block 3 DRAM space setting
		0: Do not use this block as a DRAM space
		1: Use this block as a DRAM space
1	B3WM	Block 3 wait mode
		0: Fixed wait state insertion
		1: Handshaking
2	B3BM	Block 3 bus mode
		0: Synchronous mode (synchronized with SYSCLK)
		1: Asynchronous mode (synchronized with MCLK)
5	B3BW	Block 3 bus width
		0: 16 bits 1: 32 bits
6	B3CAS	Block 3 DRAM byte specification
		0: Specified by WE3 to 0
		1: Specified by CAS3 to 0
8	B3WC0	Number of block 3 wait state insertions (LSB)
9	B3WC1	Number of block 3 wait state insertions
10	B3WC2.	Number of block 3 wait state insertions
11	B3WC3	Number of block 3 wait state insertions
12	B3WC4	Number of block 3 wait state insertions (MSB)

After reset mode is released, block 3 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. When using block 3 as a DRAM space, the number of wait states is as indicated by the setting of the WC bit in the DRAM control register. However, even when using block 3 as a DRAM space, be sure not to set a prohibited value in B3WC4 to 0.

> **!** The number of wait states that are inserted is the same as set in bits B0WC4 to 0. Refer to the description of memory control register 0 (MEMCTR0).

5.6.2.5   Memory Control Register 4

Register symbol: MEMCTR4

Address: x' 32000028

Purpose: Sets the bus mode and the number of wait states to be inserted for external memory space block 4.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	B4 WC4	B4 WC3	B4 WC2	B4 WC1	B4 WC0	-	B4 CAS	B4 BW	-	-	B4 BM	B4 WM	B4 DRAM
When reset	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0
Access	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	B4DRAM	Block 4 DRAM space setting 0: Do not use this block as a DRAM space 1: Use this block as a DRAM space
1	B4WM	Block 4 wait mode 0: Fixed wait state insertion 1: Handshaking
2	B4BM	Block 4 bus mode 0: Synchronous mode (synchronized with SYSCLK) 1: Asynchronous mode (synchronized with MCLK)
5	B4BW	Block 4 bus width 0: 16 bits 1: 32 bits
6	B4CAS	Block 4 DRAM byte specification 0: Specified by WE3 to 0 1: Specified by CAS3 to 0
8	B4WC0	Number of block 4 wait state insertions (LSB)
9	B4WC1	Number of block 4 wait state insertions
10	B4WC2	Number of block 4 wait state insertions
11	B4WC3	Number of block 4 wait state insertions
12	B4WC4	Number of block 4 wait state insertions (MSB)

After reset mode is released, block 4 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. When using block 4 as a DRAM space, the number of wait states is as indicated by the setting of the WC bit in the DRAM control register. However, even when using block 4 as a DRAM space, be sure not to set a prohibited value in B4WC4 to 0.

> **!** The number of wait states that are inserted is the same as set in bits B0WC4 to 0. Refer to the description of memory control register 0 (MEMCTR0).

5.6.2.6 Memory Control Register 5

Register symbol: MEMCTR5

Address: x' 3200002A

Purpose: Sets the wait mode, bus mode and bus width for external memory space block 5.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	B5 BW	-	-	B5 BM	B5 WM	-
When reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R

Bit No.	Bit name	Description
1	B5WM	Block 5 wait mode
		0: Fixed wait state insertion
		1: Handshaking
2	B5BM	Block 5 bus mode
		0: Synchronous mode (synchronized with SYSCLK)
		1: Asynchronous mode (synchronized with MCLK)
5	B5BW	Block 5 bus width
		0: 16 bits 1: 32 bits

The number of wait states for block 5 is indicated by the value set in bits B1WC4 to 0 in memory control register 1. After reset mode is released, block 5 is set to synchronous mode with 31 wait states, and the bus width is 32 bits. Block 5 cannot be used as a DRAM space.

5.6.2.7 Memory Control Register 6

Register symbol: MEMCTR6

Address: x' 3200002C

Purpose: Sets the wait mode and bus width for external memory space block 6.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	B6 BW	-	-	-	B6 WM	-
When reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R

Bit No.	Bit name	Description
1	B6WM	Block 6 wait mode
		0: Fixed wait state insertion
		1: Handshaking
5	B6BW	Block 6 bus width
		0: 16 bits 1: 32 bits

The number of wait states for block 6 is indicated by the value set in bits B2WC4 to 0 in memory control register 2. After reset mode is released, block 6 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. Block 6 cannot be used in asynchronous mode or as a DRAM space.

5.6.2.8 Memory Control Register 7

Register symbol: MEMCTR7

Address: x' 3200002E

Purpose: Sets the wait mode and bus width for external memory space block 7.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	B7 BW	-	-	-	B7 WM	-
When reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R

Bit No.	Bit name	Description
1	B7WM	Block 7 wait mode
		0: Fixed wait state insertion
		1: Handshaking
5	B7BW	Block 7 bus width
		0: 16 bits 1: 32 bits

The number of wait states for block 7 is indicated by the value set in bits B3WC4 to 0 in memory control register 3. After reset mode is released, block 7 is set to synchronous mode with 15 wait states, and the bus width is 32 bits. Block 7 cannot be used in asynchronous mode or as a DRAM space.

5.6.3　DRAM Control Register

5.6.3.1　DRAM Control Register

Register symbol:　　DRAMCTR

Address:　　　　　　x' 32000040

Purpose:　　　　　　Stores various DRAM mode settings when DRAM is connected.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	WC 3	WC 2	WC 1	WC 0	SIZE 1	SIZE 0	RERP	RTC	RPCP	REFE	PAGE	DRAM E
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	DRAME	DRAM control circuit enable 0: Disabled　　1: Enabled
1	PAGE	Page mode enable 0: Disabled　　1: Enabled
2	REFE	Refresh enable 0: Disabled　　1: Enabled
3	RPCP	RAS precharge cycles 0: 2 cycles (when FRQ = 0)　　1: 4 cycles (when FRQ = 0) 　　4 cycles (when FRQ = 1)　　　　6 cycles (when FRQ = 1)
4	RTC	RAS/CAS delay cycles 0: 1 cycle (when FRQ = 0)　　1: 2 cycles (when FRQ = 0) 　　2 cycles (when FRQ = 1)　　　　4 cycles (when FRQ = 1)
5	RERP	Refresh RAS cycles 0: 2 cycles (when FRQ = 0)　　1: 4 cycles (when FRQ = 0) 　　4 cycles (when FRQ = 1)　　　　6 cycles (when FRQ = 1)
6	SIZE0	DRAM size (LSB)
7	SIZE1	DRAM size (MSB) 00: For the low address, shift 8 bits to the low-order side. 01: For the low address, shift 9 bits to the low-order side. 10: For the low address, shift 10 bits to the low-order side. 11: For the low address, shift 11 bits to the low-order side.
8	WC0	DRAM wait state insertion (LSB)
9	WC1	DRAM wait state insertion
10	WC2	DRAM wait state insertion
11	WC3	DRAM wait state insertion (MSB)

The relationship between the value of WC3 to 0 and the number of wait states is shown below.

<Continued>

<Continued>

(1) When not using the in-circuit emulator (PX-ICE103002)

When FRQ = 0 (number of wait states counted by MCLK)	When FRQ = 1 (number of wait states counted by MCLK)
0000 : 0 wait states 0001 : 1 wait states 0010 : 2 wait states 0011 : 3 wait states : 0111 : 7 wait states	0000 : 0 wait states 0010 : 2 wait states 0100 : 4 wait states 0110 : 6 wait states : 1110 : 14 wait states (Setting an odd number of wait states is prohibited.)

(2) When using the in-circuit emulator (PX-ICE103002)

[1] When using the emulation function

Page mode	When FRQ = 0 (number of wait states counted by MCLK)	When FRQ = 1 (number of wait states counted by MCLK)
Used PAGE = 0	0000 : Setting prohibited 0001 : Setting prohibited 0010 : 2 wait states 0011 : 3 wait states : 0111 : 7 wait states	0000 : 0 wait states 0010 : 2 wait states 0100 : 4 wait states 0110 : 6 wait states : 1110 : 14 wait states (Setting an odd number of wait states is prohibited.)
Not used PAGE = 1	0000 : Setting prohibited 0001 : Setting prohibited 0010 : Setting prohibited 0011 : 3 wait states : 0111 : 7 wait states	0000 : Setting prohibited 0010 : 2 wait states 0100 : 4 wait states 0110 : 6 wait states : 1110 : 14 wait states (Setting an odd number of wait states is prohibited.)

[2] When using the external bus trace function

Page mode	When FRQ = 0 (number of wait states counted by MCLK)	When FRQ = 1 (number of wait states counted by MCLK)
Used PAGE = 0	0000 : 0 wait states 0001 : 1 wait states 0010 : 2 wait states 0011 : 3 wait states : 0111 : 7 wait states	0000 : 0 wait states 0010 : 2 wait states 0100 : 4 wait states 0110 : 6 wait states : 1110 : 14 wait states (Setting an odd number of wait states is prohibited.)
Not used PAGE = 1	0000 : Setting prohibited 0001 : Setting prohibited 0010 : 2 wait states 0011 : 3 wait states : 0111 : 7 wait states	

When BnDRAM of MEMCTRn is "1" and DRAME is "0", DRAM mode is not valid.

The refresh operation is not performed when REFE is "1" and DRAME is "0".

5.6.3.2 Refresh Count Register

Register symbol: REFCNT

Address: x' 32000042

Purpose: Sets the DRAM refresh interval when DRAM is connected.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	REFC 15	REFC 14	REFC 13	REFC 12	REFC 11	REFC 10	REFC 9	REFC 8	REFC 7	REFC 6	REFC 5	REFC 4	REFC 3	REFC 2	REFC 1	REFC 0
When reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0 to 15	REFC	DRAM refresh interval x'0000: 1 cycle x'FFFF: 65536 cycles (maximum value) x'0000 to x'000F: Setting prohibited

The refresh interval is the setting value multiplied by the SYSCLK cycle.

5.7   Memory Spaces

In the MN103002A/MN103002AYB, the 2 GB memory space from x'40000000 to x'BFFFFFFF is external memory space. External memory space is partitioned into 8 blocks (block 0 to block 7). When any of these blocks are accessed, various signals (such as CS) corresponding to the block in question are output. Fig. 5-7-1 shows the address format for external memory accesses, and Fig. 5-7-2 shows the memory map.
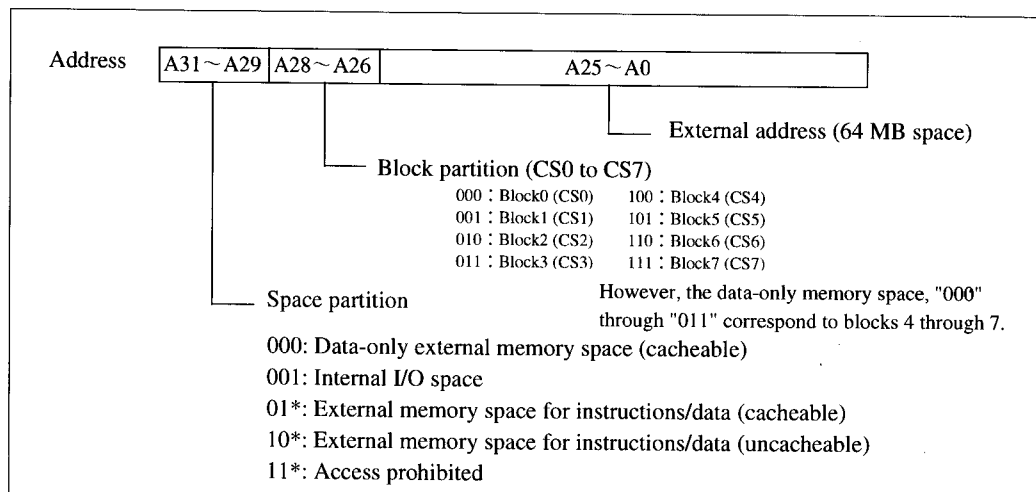
Fig. 5-7-1 Address Format When Accessing External Memory
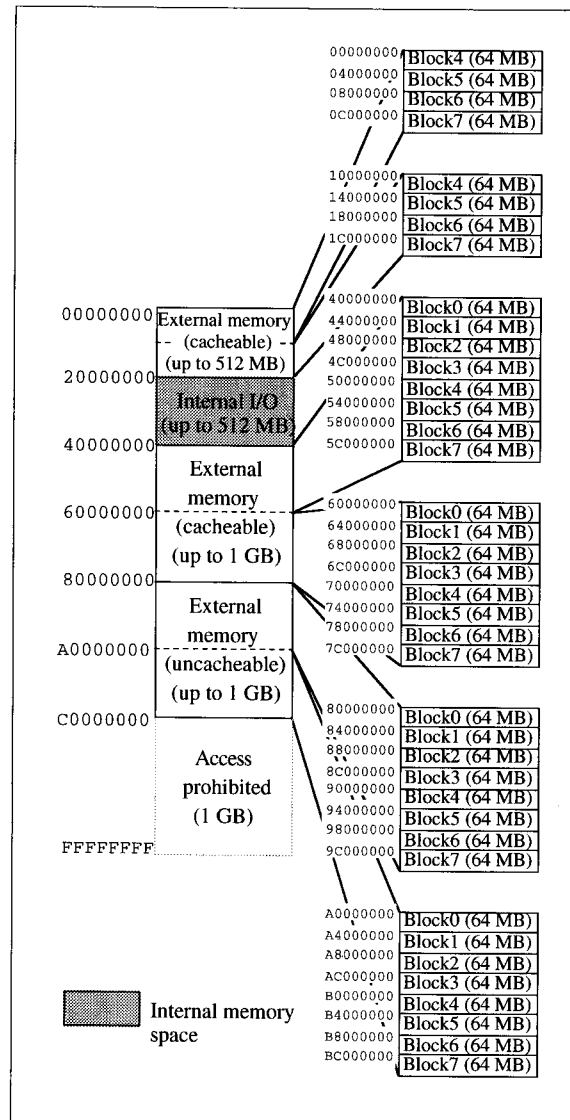
Fig. 5-7-2 Space Partitioning

Table 5-7-1 lists the characteristics of each block. The settings for each block are made in memory control registers 0 to 7 (the register number corresponds with the block number) and, when DRAM is connected, the DRAM control register.

Table 5-7-1 Characteristics of Each Block

Block	DRAM connection	Bus width	Bus access style	Number of wait states[*]	Mode	Output signal
Block 0	Not permitted	16/32	Fixed wait	1~31	Synchronous/ asynchronous	CS0
Block 1	Permitted	16/32	Fixed wait	1~31	Synchronous/ asynchronous	CS1 or RAS1
Block 2	Permitted	16/32	Fixed wait/handshake	1~31	Synchronous/ asynchronous	CS2 or RAS2
Block 3	Permitted	16/32	Fixed wait/handshake	1~31	Synchronous/ asynchronous	CS3 or RAS3
Block 4	Permitted	16/32	Fixed wait/handshake	1~31	Synchronous/ asynchronous	CS4 or RAS4
Block 5	Not permitted	16/32	Fixed wait/handshake	Same as block 1	Synchronous/ asynchronous	CS5
Block 6	Not permitted	16/32	Fixed/handshake	Same as block 2	Synchronous/	CS6
Block 7	Not permitted	16/32	Fixed/handshake	Same as block 3	Synchronous/	CS7

(*) When FRQ is "1", the number of wait states ranges from 3 to 31. When DRAM is connected, the number of wait states is determined by the setting of the WC bits in the DRAM control register.

> **!** For details on the number of wait states, refer to the description of the B0WC bits in memory control register 0 in section 5.6, "Description of Registers."

Table 5-7-2 shows examples of devices that can be connected to each block. Blocks 1 and 5 normally share the same number of wait states. Although blocks 3 and 7 normally share the same number of wait states, if DRAM is connected to block 3, the number of wait states are set independently, because the number of wait states for block 3 becomes the number set by the WC bits in the DRAM control register, while the number of wait states for block 7 becomes the number set by the B3WC bits in memory control register 3. In addition, because block 6 uses the same number of wait states as block 2, they are used via handshaking. Note that handshaking can only be used in synchronous mode.

Table 5-7-2 Examples of Devices That Can Be Connected Each Block

Block	Connected device	Bus access style	Mode	Output signal
Block 0	ROM	Fixed wait (MEMCTR0)	Asynchronous	CS0
Block 1	SRAM1	Fixed wait (MEMCTR1)	Asynchronous	CS1
Block 2	SRAM2	Fixed wait (MEMCTR2)	Synchronous	CS2
Block 3	DRAM1	Fixed wait (DRAMCTR)	—	RAS3
Block 4	DRAM2	Fixed wait (DRAMCTR)	—	RAS4
Block 5	I/O device 1	Fixed wait (MEMCTR5)	Asynchronous	CS5
Block 6	I/O device 2	Handshake	Synchronous	CS6
Block 7	I/O device 3	Handshake	Synchronous	CS7

5.8 Description of Operation

5.8.1 Number of Basic Bus Cycles

The frequency of the CPU operation clock (MCLK) can be selected through the setting of the FRQS external input pin. If FRQS is high, the frequency of the CPU operation clock is four times the input frequency (which is the same frequency as SYSCLK); if FRQS is low, the frequency of the CPU operation clock is twice the input frequency. Furthermore, the frequency of the internal I/O section operation clock (IOCLK) is also determined by the setting of the FRQS external input pin. If FRQS is high, the frequency of the internal I/O section operation clock is equal to the input frequency; if FRQS is low, the frequency of the internal I/O section operation clock is one-half the input frequency.

Table 5-8-1 shows examples of the number of CPU cycles that are needed for various types of CPU access when FRQS is high and when FRQS is low.

Table 5-8-1 Relationship Between the Clock Frequency and the Number of Cycles (CPU Cycles) Required for Access

Destination of access			FRQS = H MCLK 66 MHz SYSCLK 16.5 MHz IOCLK 16.5 MHz	FRQS = L MCLK 66 MHz SYSCLK 33 MHz IOCLK 16.5 MHz
Instruction cache	Instruction read	Hit	2	2
		Miss	Number of external memory cycles x 2 + 2	Number of external memory cycles x 2 + 2
Data cache	Read	Hit	1	1
		Miss	Number of external memory cycles + 2	Number of external memory cycles + 2
	Write	Hit	1	1
		Miss	Number of external memory cycles + 2	Number of external memory cycles + 2
Control register in CPU	Read/write		Read: 3/write: 2	Read: 3/write: 2
Control register in BC	Read/write		Read: 3/write: 2	Read: 3/write: 2
Internal I/O	Read	Synchronous[*2]	Number of I/O bus cycles + 2 to 5	Number of I/O bus cycles + 2 to 3
		Asynchronous	Number of I/O bus cycles + 2	Number of I/O bus cycles + 2
	Write[*1]	Synchronous[*2]	Number of I/O bus cycles + 1 to 4	Number of I/O bus cycles + 1 to 2
		Asynchronous	Number of I/O bus cycles + 1	Number of I/O bus cycles + 1
External memory	Read	Synchronous[*3]	Number of EX bus cycles + 0 to 3	Number of EX bus cycles + 0 to 2
		Asynchronous	Number of EX bus cycles + 0 to 4	Number of EX bus cycles + 0 to 2
	Write[*1]	Synchronous[*3]	Number of EX bus cycles + 0 to 3	Number of EX bus cycles + 0 to 2
		Asynchronous	Number of EX bus cycles + 0 to 4	Number of EX bus cycles + 0 to 2

[*1] If the store buffer is used, all writes to internal I/O and external memory are executed in one cycle.

[*2] In synchronous mode, a wait of a maximum of three cycles for synchronization is generated.

[*3] In synchronous mode, when the MCLK frequency is four times the SYSCLK frequency, a wait of a maximum of three cycles for synchronization is generated; if the MCLK frequency is twice the SYSCLK frequency, a wait of a maximum of one cycle for synchronization is generated.

5.8.2　Store Buffer

The bus controller has one store buffer (with a 32-bit data width) built in, and is used to avoid a time penalty when conducting a store operation in internal I/O or external memory. The CPU store operation is completed by storing the address, data, and access size in the store buffer, and is executed with no wait states. Writes from the store buffer to internal I/O or external memory are conducted in parallel with subsequent CPU operations. However, if there is a load request or a store request directed at internal I/O or external memory from the CPU before the write from the store buffer is completed, the execution of that request is delayed.

5.8.3　Accessing the Internal I/O Space

5.8.3.1　Synchronous Mode Access

Accesses to the internal I/O space (I/O register) are performed through the I/O bus, with the bus controller controlling the interface for read/write requests from the CPU. In synchronous mode, accesses between the bus controller and the internal I/O space are executed in synchronization with IOCLK and with the number of wait states (0 or 1) set in the I/O bus control register. Fig. 5-8-1 shows the timing chart when accessing the internal I/O space with no wait states, and Fig. 5-8-2 shows the timing chart when accessing the internal I/O space with one wait state.
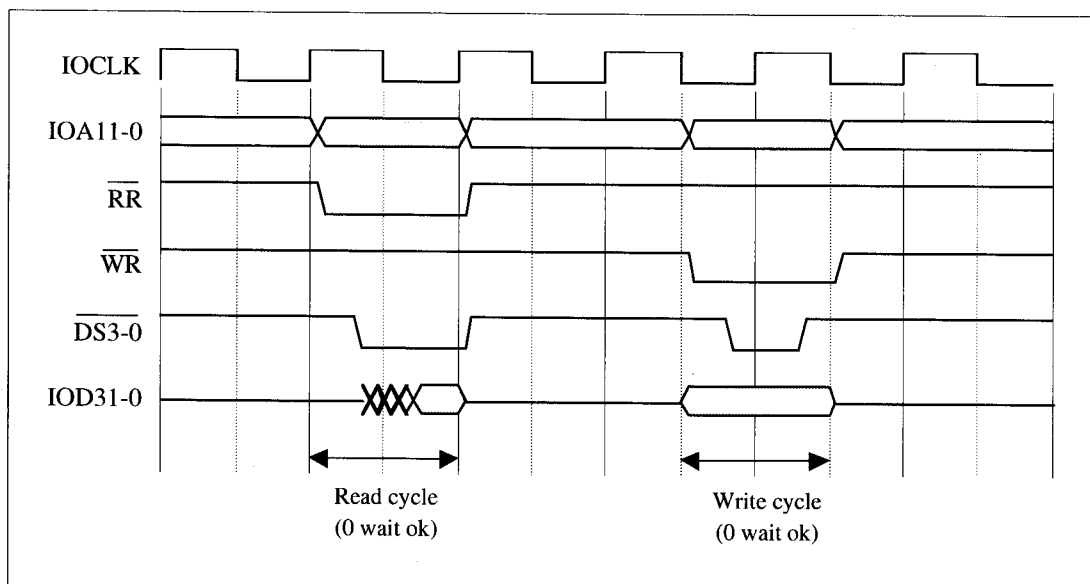
Fig. 5-8-1　Internal I/O Space Access (Synchronous Mode: No Wait OK)
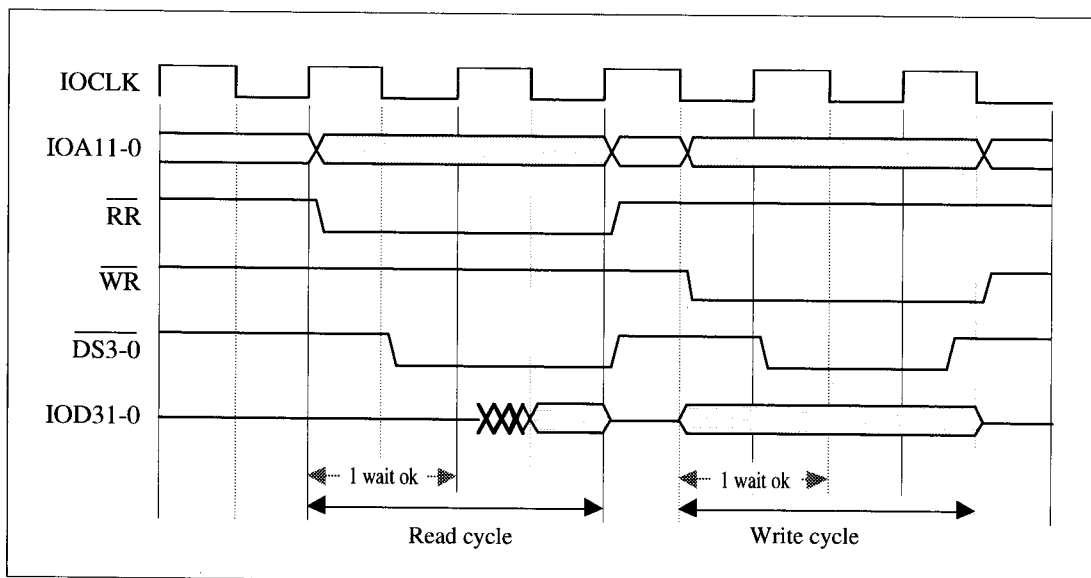
Fig. 5-8-2 Internal I/O Space Access (Synchronous Mode: 1 Wait OK)

During a read, the address (IOA11 to 0) and the read request signal (RR) are output in synchronization with the rising edge of IOCLK. Then, after 1/4 of an IOCLK cycle (1MCLK) in the no wait states example or after 1/2 IOCLK (2MCLK) cycle in the one wait state example, the data strobe signals (DS3 to 0) are asserted, and the I/O side begins to drive the data on the data bus. During a write, the address (IOA11 to 0) and the write request signal (WR) are output in synchronization with the falling edge of IOCLK. Then, after 1/4 of an IOCLK cycle (1MCLK) in the no wait states example or after 1/2 IOCLK (2MCLK) cycle in the one wait state example, the data strobe signals (DS3 to 0) are asserted, and are then negated 1/4 of an IOCLK cycle (1MCLK) before the end of the I/O access cycle. The write is performed at the rising edge of the DS3 to 0 signals.

Note that in the case of a normal internal I/O circuit, the bus cycle is completed with the number of wait states (0 or 1 IOCLK cycles) set by the IOWC bits in the I/O bus control register, but in the case of the I/O port only, the bus cycle is completed at the number of wait states (0 to 3 IOCLK cycles) set by the IOPWC bits in the I/O bus control register.

5.8.3.2   Asynchronous Mode Access

Asynchronous mode access is provided in order to permit high-speed access of the internal I/O space.

In order to perform an asynchronous mode access, set the IOBM bit in the I/O bus control register (IOCTR). Once the IOBM bit is set, all I/O accesses are conducted in synchronization with the MCLK; because there is no wait for synchronization with IOCLK in response to CPU requests (a wait of up to three MCLK cycles), high-speed access becomes possible.

However, during serial reception, etc., if the data changes during the read cycle, the read data is not guaranteed; therefore, it is necessary either to read the data after serial reception is completed, or else to change the mode to synchronous mode before reading an internal I/O circuit where there is a possibility of the data changing. In addition, when writing in asynchronous mode, operation is not guaranteed due to a problem with the mode switching timing; therefore, do not use asynchronous mode except with certain I/O circuits.

The number of wait states in asynchronous mode is set through the IOWC bits in the I/O bus control register. Fig. 5-8-3 shows the timing for an internal I/O access in asynchronous mode when the number of wait states is set to "3".

> **Setting synchronous mode with no wait states is recommended except when otherwise necessary.**
> **Operation is not guaranteed for accesses in asynchronous mode. For details, refer to the specifications for the peripheral circuits.**
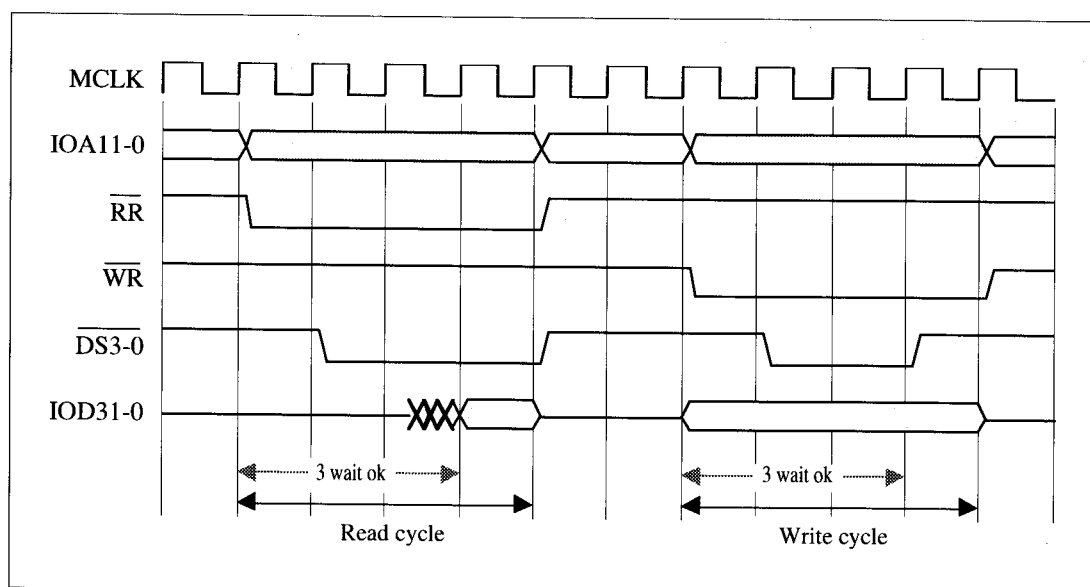
Fig. 5-8-3   Internal I/O Space Access (Asynchronous Mode with Three Wait OK)

During a read, the address (IOA11 to 0) and the read request signal (RR) are output in synchronization with the rising edge of MCLK. After one MCLK cycle, the data strobe signals (DS3 to 0) are asserted, and the I/O side begins to drive the data on the data bus. During a write, the address (IOA11 to 0) and the write request signal (WR) are output in synchronization with the rising edge of MCLK. After one MCLK cycle, the data strobe signals (DS3 to 0) are asserted, and are then negated one MCLK cycle before the end of the I/O access cycle. The write is performed at the rising edge of the DS3 to 0 signals.

Note that in the case of a normal internal I/O circuit, the bus cycle is completed at the number of wait states (0 to 7 MCLK cycles) set by the IOWC bits in the I/O bus control register, but in the case of the I/O port only, the bus cycle is completed at the number of wait states (0 to 15 MCLK cycles) set by the IOPWC bits in the I/O bus control register.

5.8.4 External Memory Space Access (non-DRAM Spaces)

5.8.4.1 Bus Synchronous Modes

32-bit bus synchronous mode

Setting of the various parameters for external memory access is performed in memory control registers 0 to 7, corresponding to each block. In synchronous mode, the bus access is initiated in synchronization with SYSCLK. When fixed wait state insertion is specified, the bus access ends after the number of wait states set in the memory control register; if handshaking is specified, the bus access ends in the cycle following the cycle in which the data acknowledge signal (DK) is asserted.

Figs. 5-8-4 and 5-8-5 are timing charts for accesses with a 32-bit bus in synchronous mode with fixed wait state insertion. In Fig. 5-8-4, one wait state is inserted, while in Fig. 5-8-5, two wait states are inserted. During a read, the read enable signal (RE) is asserted 1/2 of a SYSCLK cycle after the start of the bus cycle, and is negated 1/2 of a SYSCLK cycle before the end of the bus cycle. During a write, the write enable signals (WE3 to 0: corresponding to the byte being written) are asserted 1/2 of a SYSCLK cycle after the start of the bus cycle and are negated 1/2 of a SYSCLK cycle before the end of the bus cycle. Therefore, the minimum number of wait states in the bus cycle is one.

When writing byte 0, WE0 is asserted and the data is output on D7 to 0. When writing byte 1, WE1 is asserted and the data is output on D15 to 8. When writing byte 2, WE2 is asserted and the data is output on D23 to 16. When writing byte 3, WE3 is asserted and the data is output on D31 to 24.
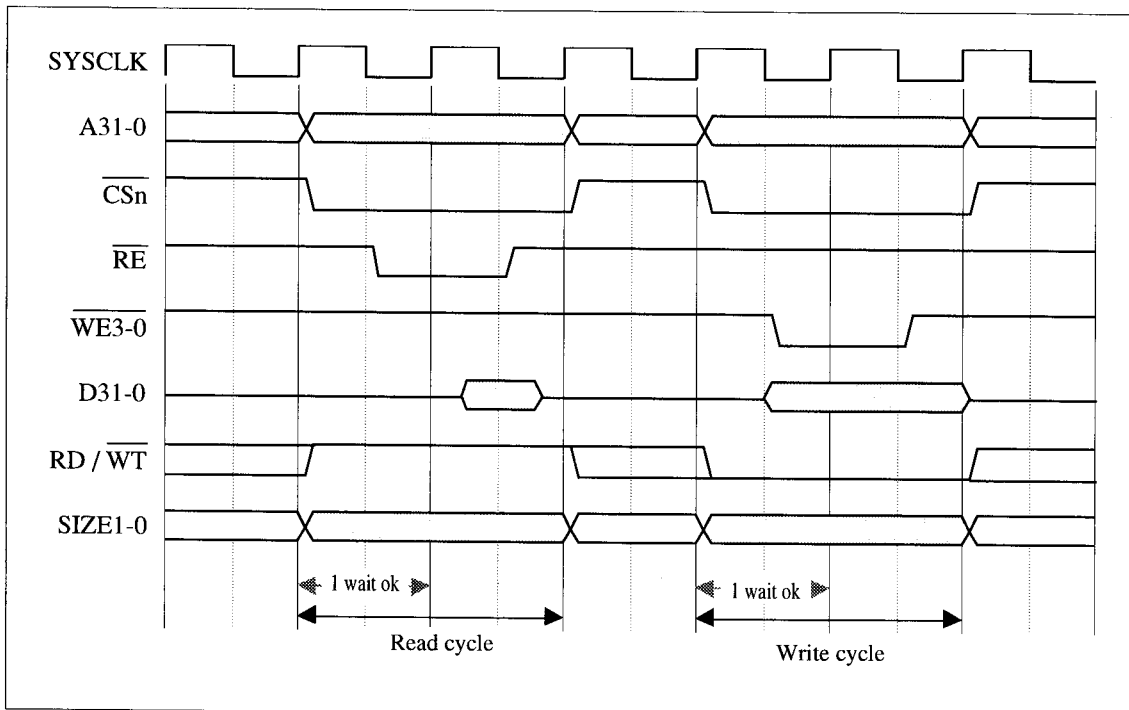
Fig. 5-8-4 Timing for Access in 32-bit Bus Synchronous Mode with Fixed Wait State Insertion (One Wait OK)
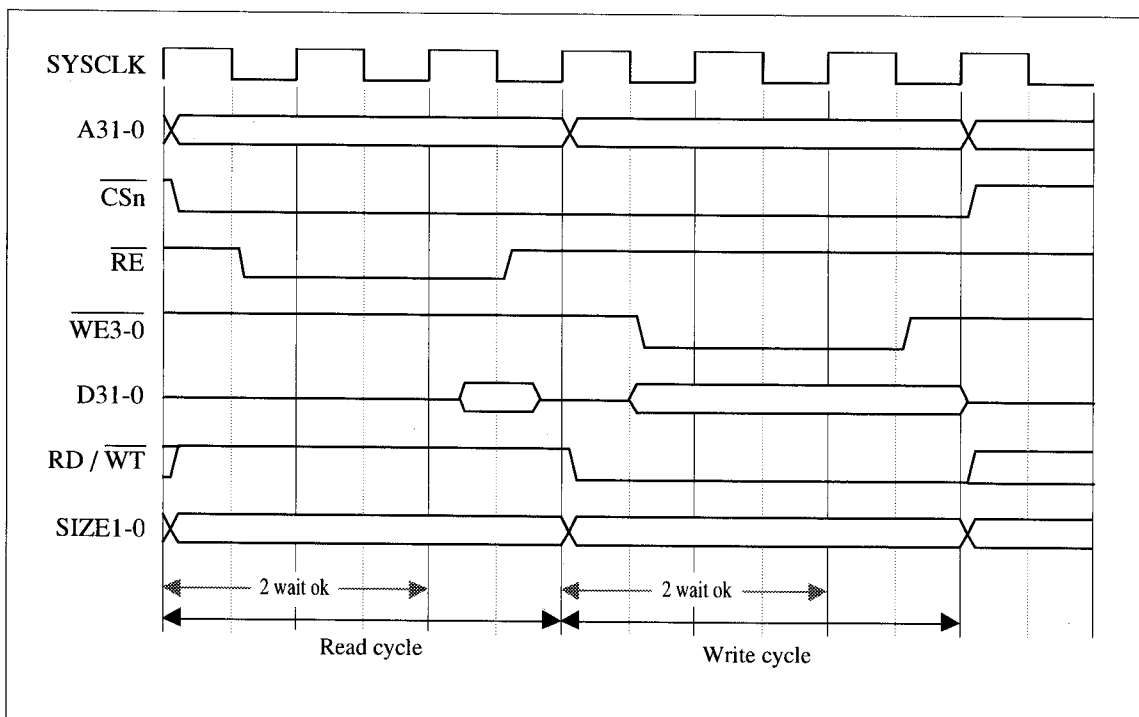
Fig. 5-8-5 Timing for Access in 32-bit Bus Synchronous Mode with Fixed Wait State Insertion (Two Wait OK)

Fig. 5-8-6 is a timing chart for an access with a 32-bit bus and handshaking specified. When handshaking is specified, the data acknowledge signal (DK) is sampled at the rising edge of SYSCLK, and if it is asserted the bus access ends at the next cycle. Note that when the DK signal is found to have been asserted, it is not sampled in the next cycle. Bus access termination by handshaking is valid only in synchronous mode.
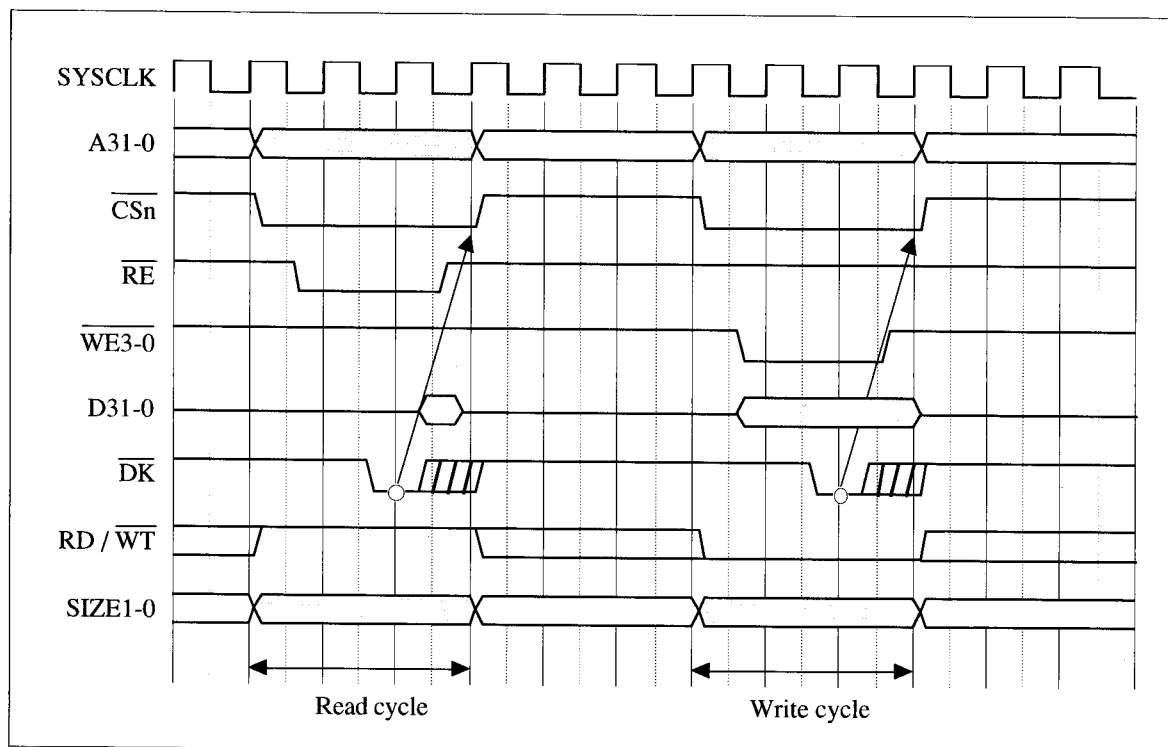
Fig. 5-8-6 Timing for Bus Access by 32-bit Bus Handshaking

16-bit bus synchronous mode

16-bit bus mode can be entered by inputting "0" to the BMODE pin for block 0, or by setting "0" in the BnBW 1 to 0 bit in the memory control registers for blocks 1 to 7.

Figs. 5-8-7 and 5-8-8 are timing charts for accesses with a 16-bit bus in synchronous mode with fixed wait state insertion.

In Fig. 5-8-7, one wait state is inserted, while in Fig. 5-8-8, two wait states are inserted.

In 16-bit bus mode, a word access (32 bits) is performed through two external accesses, one for the lower half-word (A[1] = 0) and one for the upper half-word (A[1] = 1). In the case of a half-word access (16 bits), or a byte access (8 bits), the access is performed through only one external access of the corresponding address. The most significant 16 bits (D31 to 16) of the data bus are used.

When writing byte 0 and byte 2, WE2 is asserted and the data is output on D23 to 16. When writing byte 1 and byte 3, WE3 is asserted and the data is output on D31 to 24.
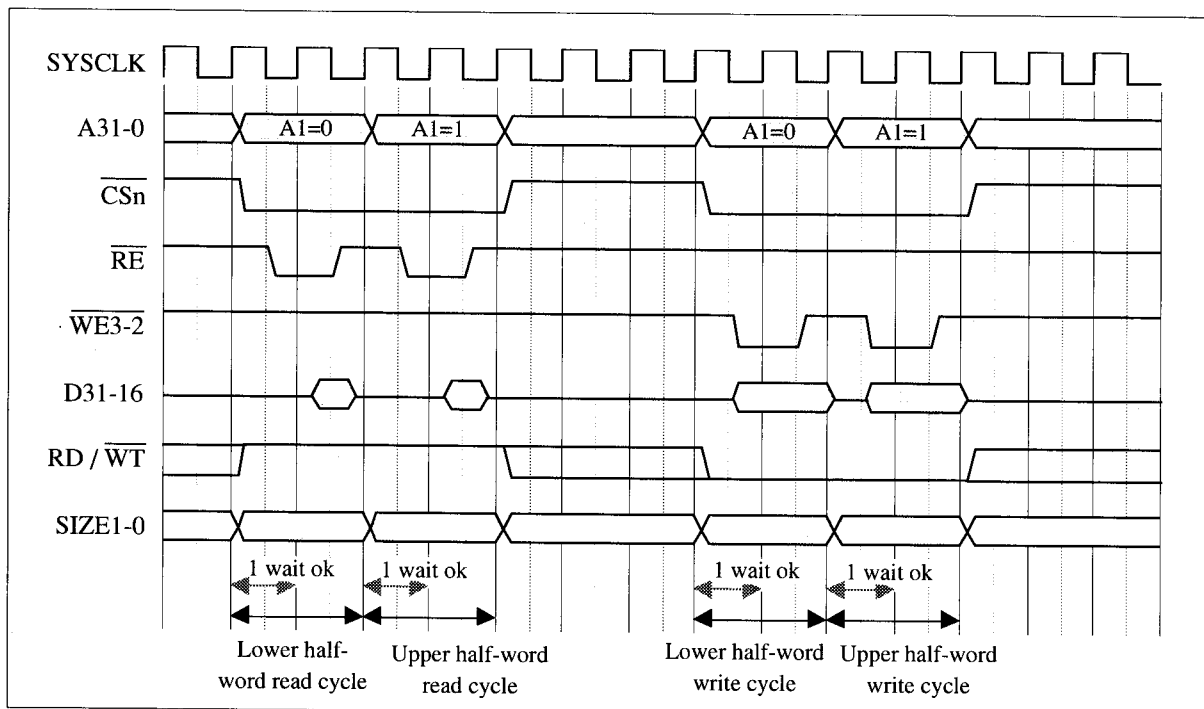
Fig. 5-8-7 Timing for Access in 16-bit Bus Synchronous Mode with Fixed Wait State Insertion (One Wait OK)
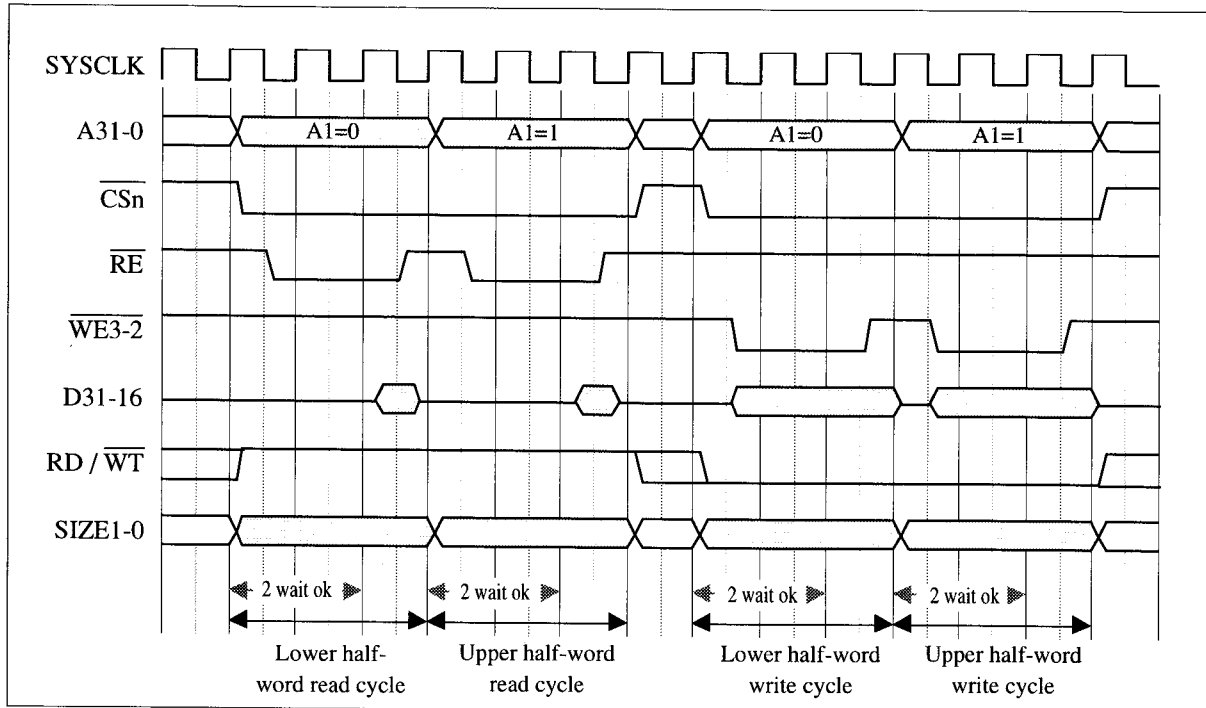
Fig. 5-8-8 Timing for Access in 16-bit Bus Synchronous Mode with Fixed Wait State Insertion (Two Wait OK)

Fig. 5-8-9 is a timing chart for an access with a 16-bit bus and handshaking specified. When handshaking is specified, the data acknowledge signal (DK) is sampled at the rising edge of SYSCLK, and if it is asserted the bus access ends at the next cycle. Note that when the DK signal is found to have been asserted, it is not sampled in the next cycle. Bus access termination by handshaking is valid only in synchronous mode.
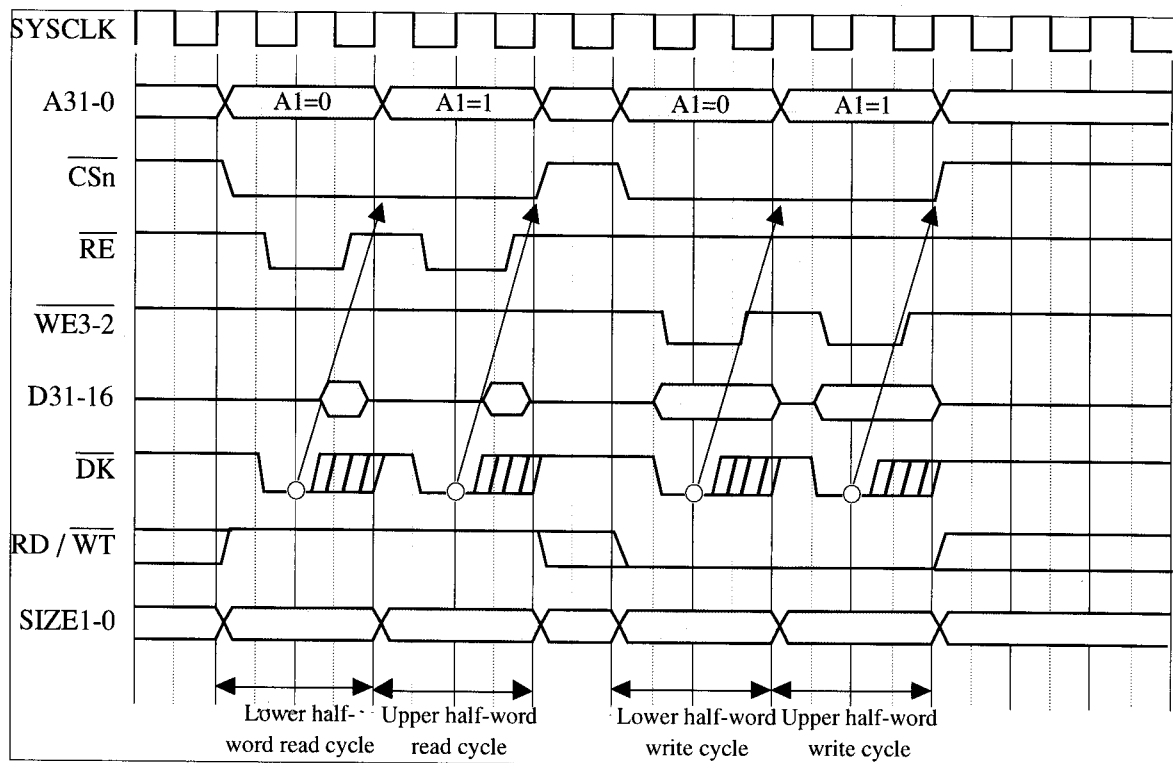
Fig. 5-8-9 Timing for Bus Access by 16-bit Bus Handshaking

5.8.4.2 Asynchronous Modes

32-bit bus asynchronous mode (when FRQ = 0)

Asynchronous mode is used for accessing external memory at high speed; the address signals, CS signals, etc., are output asynchronously with the external clock but synchronously with the internal MCLK. In asynchronous mode, accesses are all by fixed wait state insertion, not by handshaking.

Fig. 5-8-10 is a timing chart for an access with a 32-bit bus in asynchronous mode. During a read, the RE signal is asserted 1/2 of an MCLK cycle after the start of the bus cycle, and is negated 1/2 of an MCLK cycle before the end of the bus cycle. During a write, the WE signal is asserted 1/2 of an MCLK cycle after the start of the bus cycle and is negated 1/2 of an MCLK cycle before the end of the bus cycle. The minimum number of wait states for the bus cycle when FRQ is "0" is one wait state.

When writing byte 0, WE0 is asserted and the data is output on D7 to 0. When writing byte 1, WE1 is asserted and the data is output on D15 to 8. When writing byte 2, WE2 is asserted and the data is output on D23 to 16. When writing byte 3, WE3 is asserted and the data is output on D31 to 24.
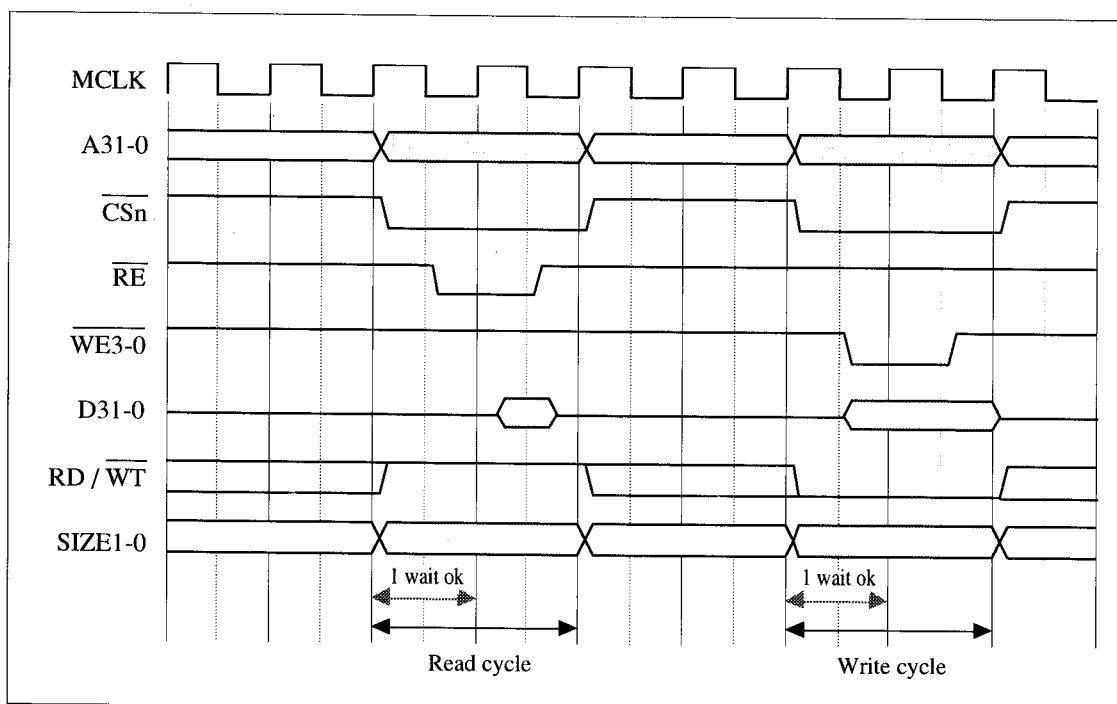
Fig. 5-8-10 Timing for Access in Asynchronous Mode with 32-bit Bus (FRQ = 0, One Wait OK)

32-bit bus asynchronous mode (when FRQ = 1)

Fig. 5-8-11 is a timing chart for an access with a 32-bit bus in asynchronous mode. During a read, the RE signal is asserted one MCLK cycle after the start of the bus cycle, and is negated one MCLK cycle before the end of the bus cycle. During a write, the WE signal is asserted one MCLK cycle after the start of the bus cycle and is negated one MCLK cycle before the end of the bus cycle. The minimum number of wait states for the bus cycle when FRQ is "1" is three wait states.

When writing byte 0, WE0 is asserted and the data is output on D7 to 0. When writing byte 1, WE1 is asserted and the data is output on D15 to 8. When writing byte 2, WE2 is asserted and the data is output on D23 to 16. When writing byte 3, WE3 is asserted and the data is output on D31 to 24.
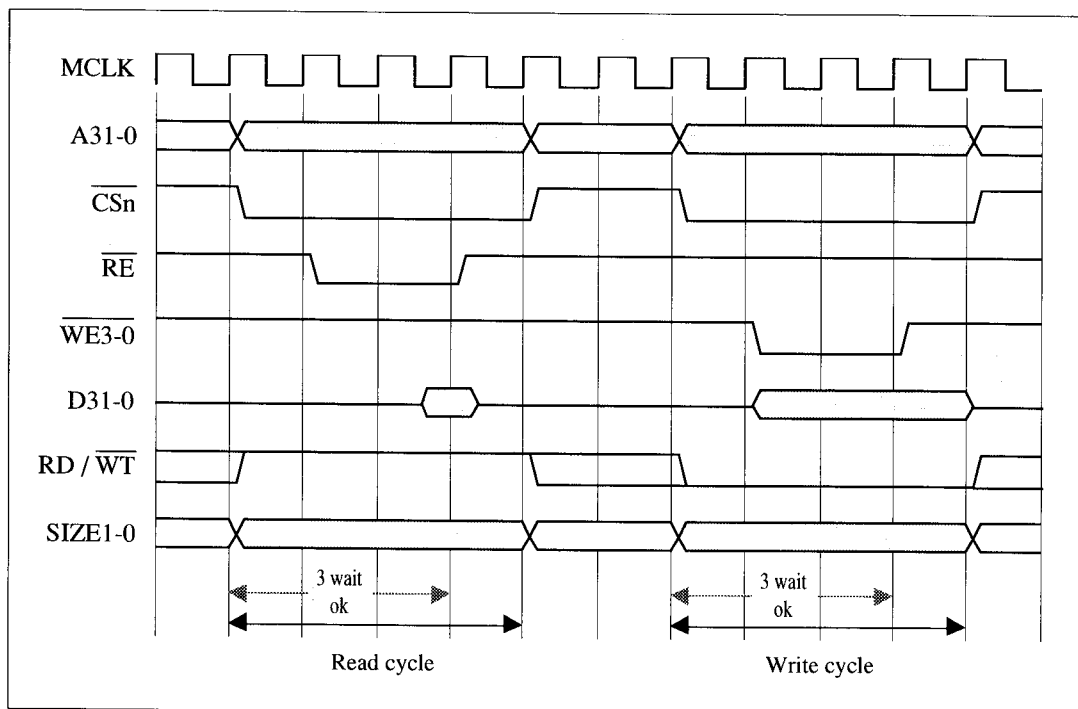
Fig. 5-8-11 Timing for Access in Asynchronous Mode with 32-bit Bus (FRQ = 1, Three Wait OK)

16-bit bus asynchronous mode

16-bit bus mode can be entered by inputting "0" to the BMODE pin for block 0, or by setting "0" in the BnBW 1 to 0 bit in the memory control registers for blocks 1 to 7. In 16-bit bus mode, a word access (32 bits) is performed through two external accesses, one for the lower half-word (A[1] = 0) and one for the upper half-word (A[1] = 1). In the case of a half-word access (16 bits), or a byte access (8 bits), the access is performed through only one external access of the corresponding address. The most significant 16 bits (D31 to 16) of the data bus are used. Fig. 5-8-12 is a timing chart for a word access with a 16-bit bus in asynchronous mode with FRQ = 0, while Fig. 5-8-13 is a timing chart for a word access with a 16-bit bus in asynchronous mode with FRQ = 1.

When writing byte 0 and byte 2, WE2 is asserted and the data is output on D23 to 16. When writing byte 1 and byte 3, WE3 is asserted and the data is output on D31 to 24.
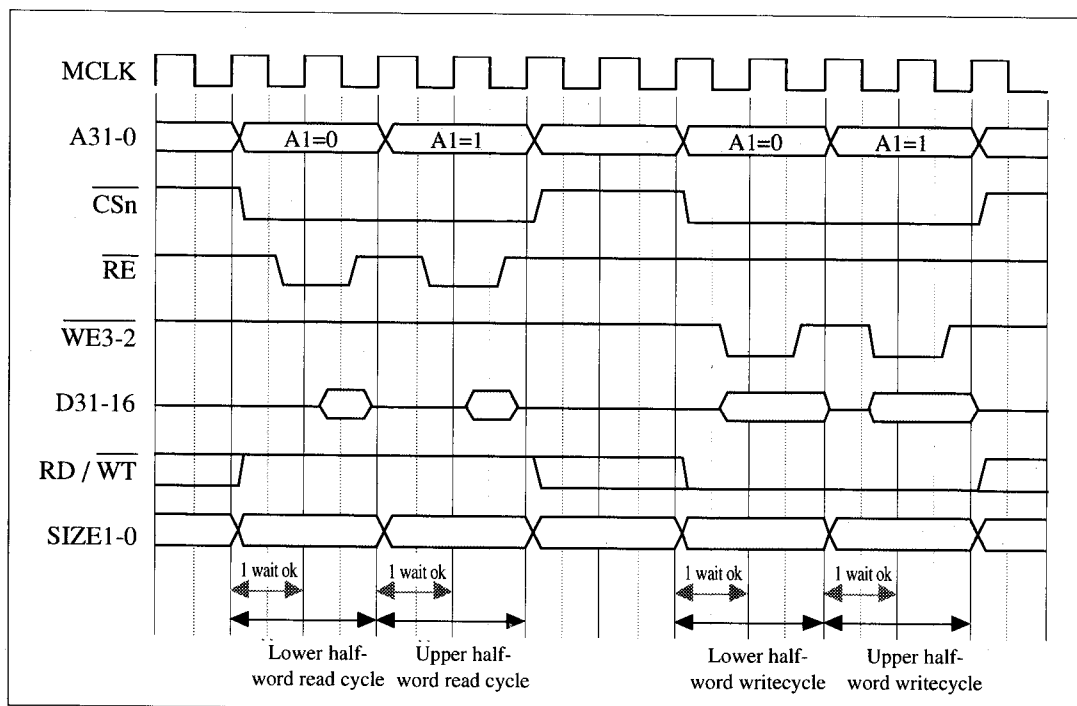
Fig. 5-8-12 Timing for Access in Asynchronous Mode with 16-bit Bus (FRQ = 0, One Wait OK)
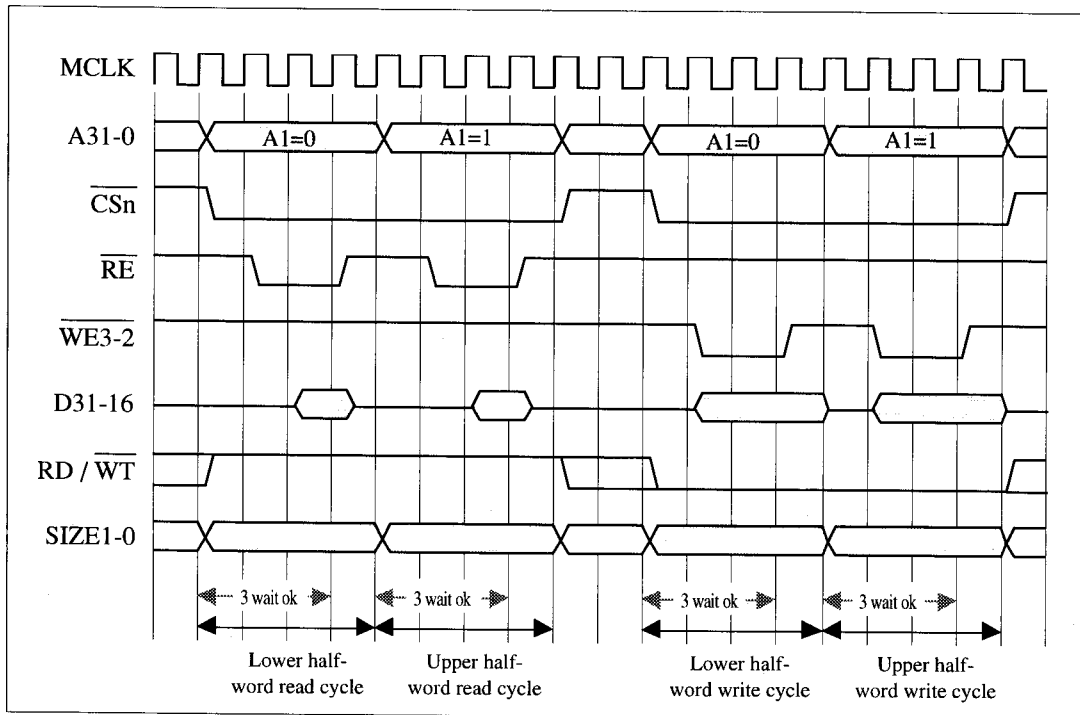
Fig. 5-8-13 Timing for Access in Asynchronous Mode with 16-bit Bus (FRQ = 1, Three Wait OK)

5.8.5 External Memory Space Access (DRAM Space)

5.8.5.1 RAS/CAS Signal Timing

Blocks 1 to 4 can be used as DRAM space by setting the BnDRAM bits in memory control registers 1 to 4. By setting the DRAME bit in the DRAM control register, it is possible to generated multiplexed address output, RAS/CAS signal output, etc.. (When BnDRAM = 1 and DRAME = 0, operation is the same as if BnDRAM = 0.) For the byte specification, output is on the CAS3 through 0 signals when the BnCAS bit is set in memory control registers 1 through 4. When the BnCAS bit is cleared, output is on the WE3 through 0 signals. (When using CAS3 through 0, WE3 is used as the WE signal; when using WE3 through 0, CAS3 is used as the CAS signal.) The DRAM bus cycle is always asynchronous with the external clock (and synchronized with MCLK). The RAS/CAS signal output timing can be set through the RTC bit, RPCP bit, and WC3 through 0 bits in the DRAM control register. Figs. 5-8-14 and Fig. 5-8-15 show the timing charts. The number of cycles is always given in terms of the number of MCLK cycles.
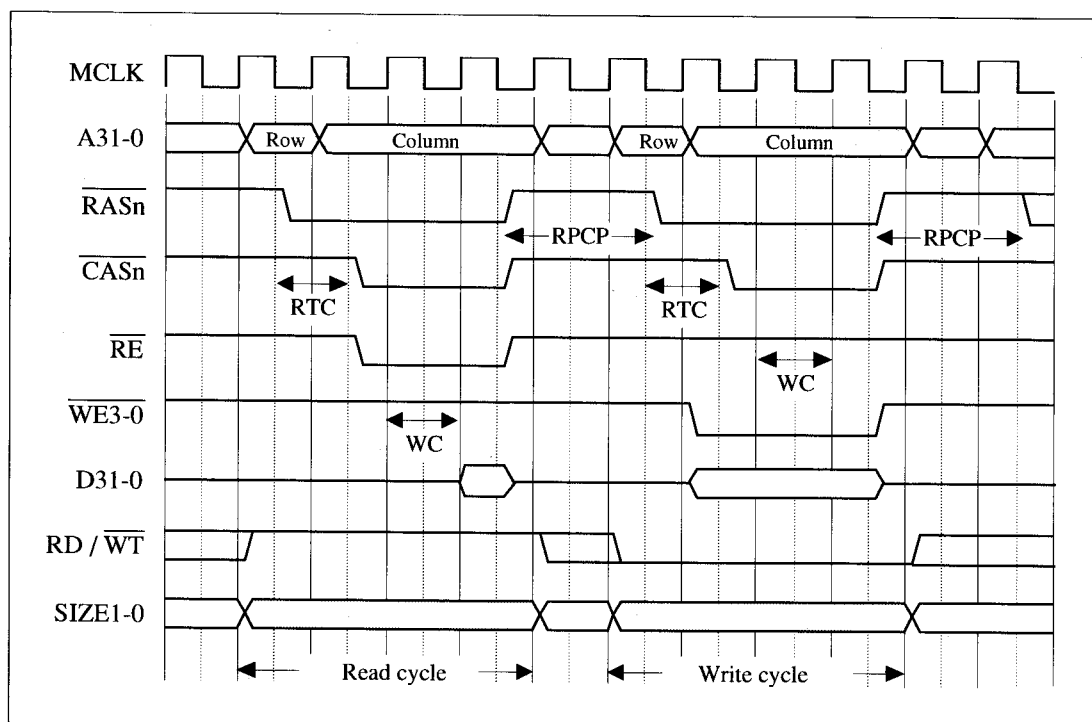
Fig. 5-8-14 RAS/CAS Signal Timing (When FRQ = 0)
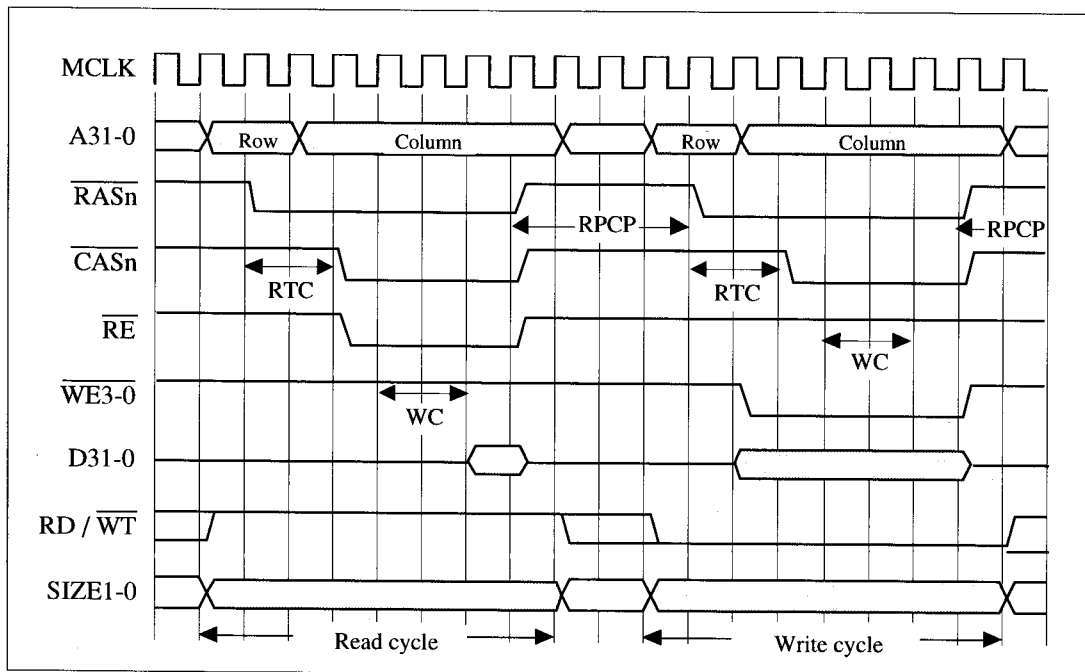
Fig. 5-8-15 RAS/CAS Signal Timing (When FRQ = 1)

5.8.5.2 Normal Mode

This section explains the timing of the 32-bit DRAM bus cycle for the basic cycle (when the DRAMCTR bits RTC = 0, RPCP = 0, PAGE = 0, and WC = 0001.)

> ! For details on the settings of the RTC, RPCP, PAGE, and WC bits, refer to
> section 5.6.3.1, "DRAM Control Register."

DRAM read/write (when FRQ = 0)

Regarding accesses to DRAM spaces, the address is shifted down and the row address is output according to the value of the SIZE1 and 0 bits in the DRAM control register. The byte specification by WE3 through 0 is set by clearing the BnCAS bit in memory control registers 1 through 3. The RASn signal ("n" corresponds to the block number) and the CAS3 signal are then output according to the values of the RTC bit, RPCP bit and the WC3 through 0 bits in the DRAM control register.

The byte specification by CAS3 through 0 is set by setting the BnCAS bit in memory control registers 1 through 3. (In this case, WE3 is used.) Fig. 5-8-16 shows the 32-bit bus timing and Fig. 5-8-17 shows the 16-bit bus timing.
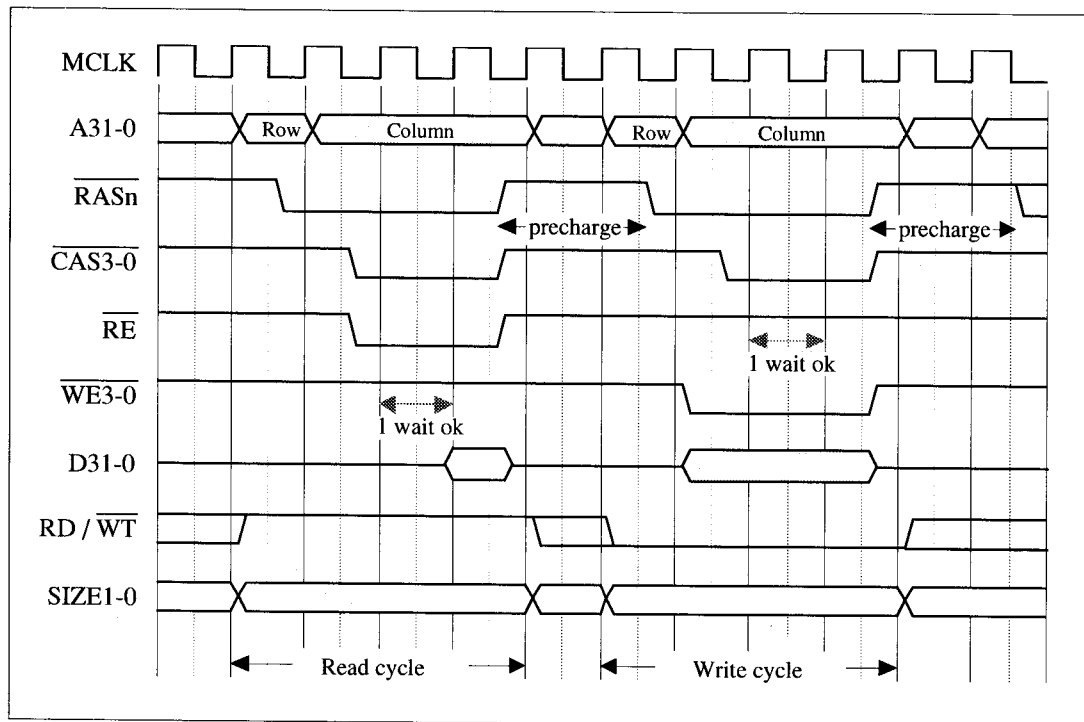
Fig. 5-8-16 32-bit DRAM Read/Write Timing (When FRQ = 0, One Wait OK)
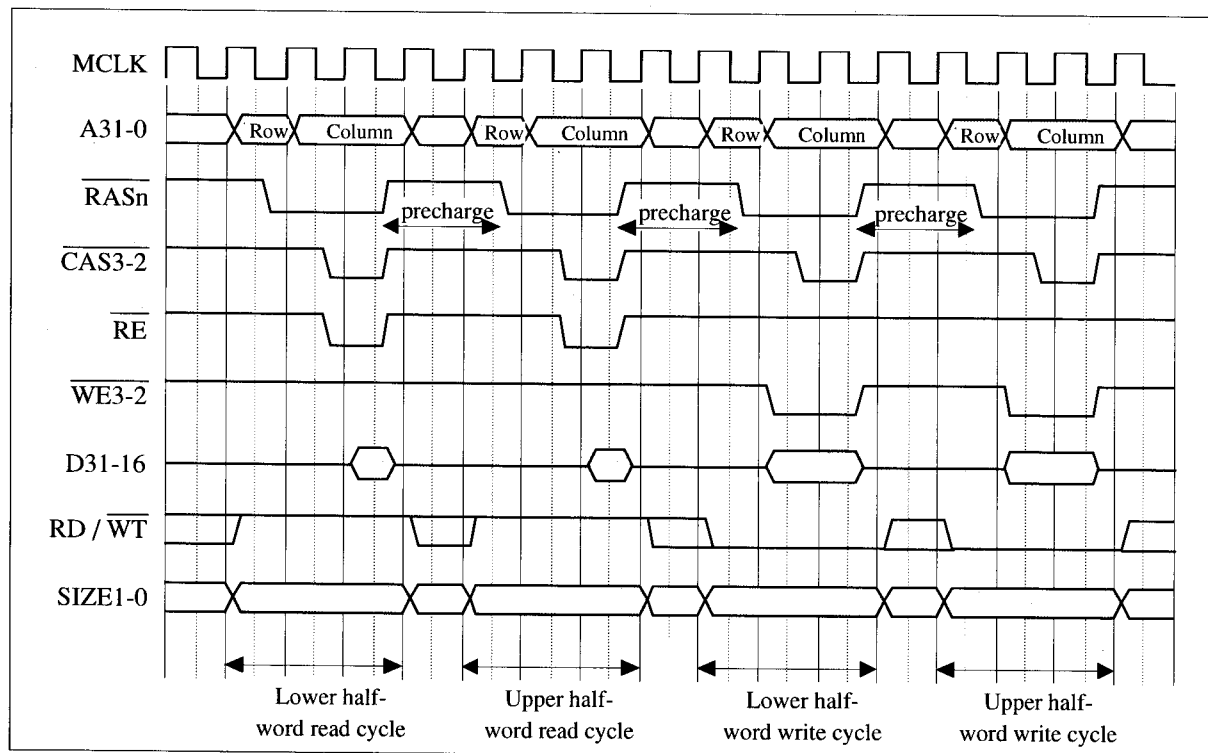
Fig. 5-8-17 16-bit DRAM Read/Write Timing (When FRQ = 0)

DRAM read/write (when FRQ = 1)

Fig. 5-8-18 shows the 32-bit bus DRAM read/write timing when FRQ = 1 and Fig. 5-8-19 shows the 16-bit bus DRAM read/write timing when FRQ = 1. Just as when FRQ = 0, the address is shifted down and the row address is output according to the value of the SIZE1 and 0 bits in the DRAM control register. The RASn signal ("n" corresponds to the block number) and the CAS signal are then asserted according to the values of the RTC bit, RPCP bit and the WC3 through 0 bits in the DRAM control register. Byte specification is the same as when FRQ = 0.
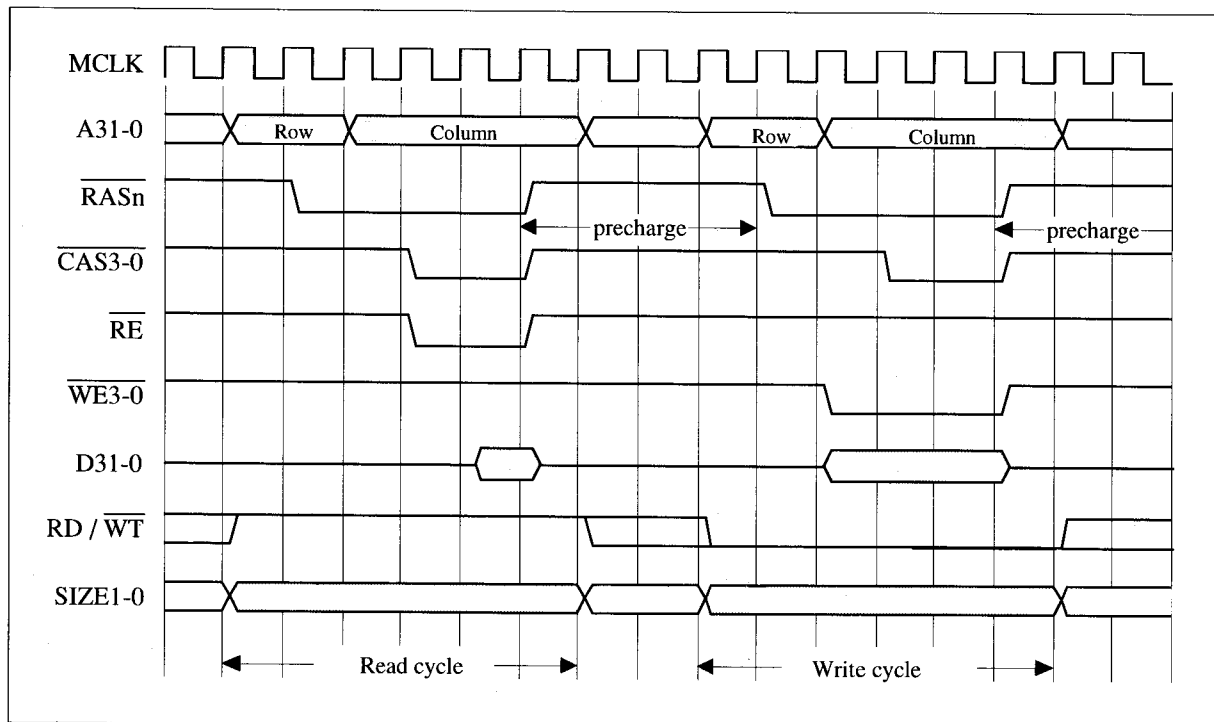
Fig. 5-8-18 32-bit DRAM Read/Write Timing (When FRQ = 1)
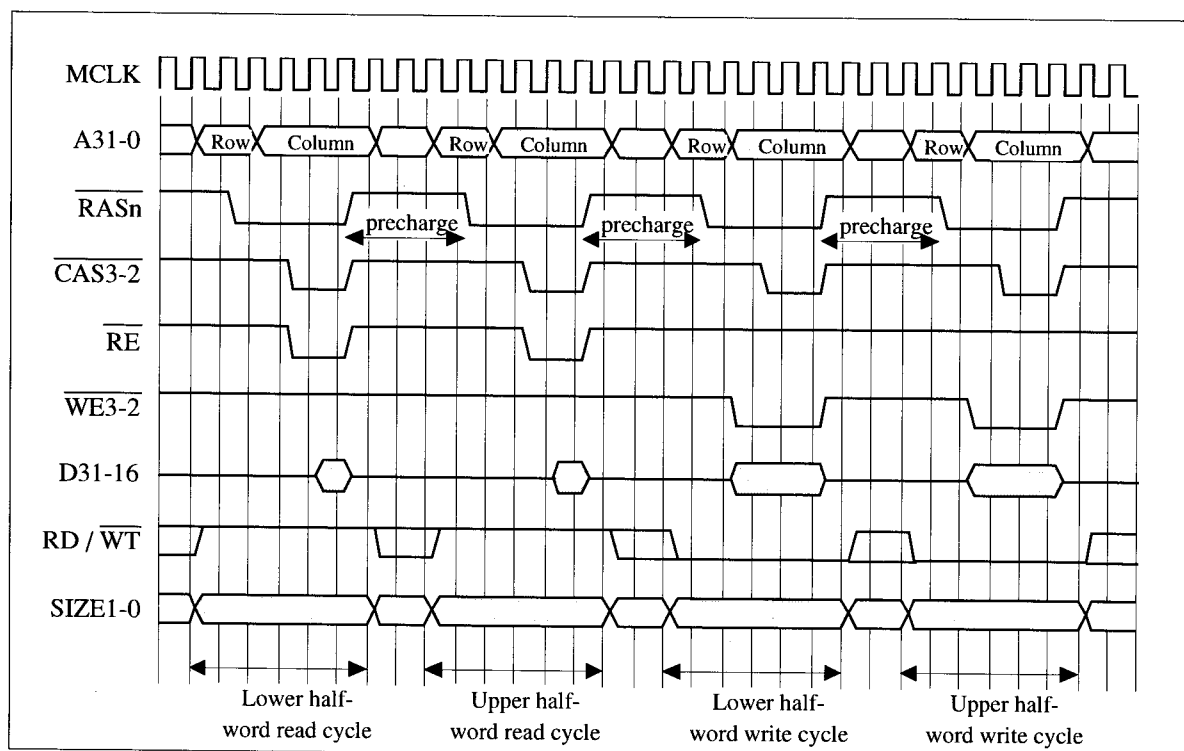
Fig. 5-8-19 16-bit DRAM Read/Write Timing (When FRQ = 1)

5.8.5.3 Page Mode

DRAM page mode (when FRQ = 0)

If the PAGE bit in the DRAM control register is set, page mode access is enabled, making high-speed access in page mode possible for consecutive accesses to DRAM.

The byte specification by WE3 through 0 is set by clearing the BnCAS bit in memory control registers 1 through 3. The RAS signal and, in the case of a write, the WE signal are negated as soon as the consecutive accesses to DRAM are completed.

The byte specification by CAS3 through 0 is set by setting the BnCAS bit in memory control registers 1 through 3. Just as in the case of byte specification by WE3 through WE0, the RAS signal and, in the case of a write, the WE signal are negated as soon as the consecutive accesses to DRAM are completed.

The address is shifted down and the row address is output according to the value of the SIZE1 and 0 bits in the DRAM control register. The RASn signal ("n" corresponds to the block number) and the CAS signal are then asserted according to the values of the RTC bit, RPCP bit and the WC3 through 0 bits in the DRAM control register.

Fig. 5-8-20 shows the page mode read timing with a 32-bit bus, and Fig. 5-8-22 shows the page mode read timing with a 16-bit bus. Fig. 5-8-21 shows the page mode write timing with a 32-bit bus, and Fig. 5-8-23 shows the page mode write timing with a 16-bit bus.
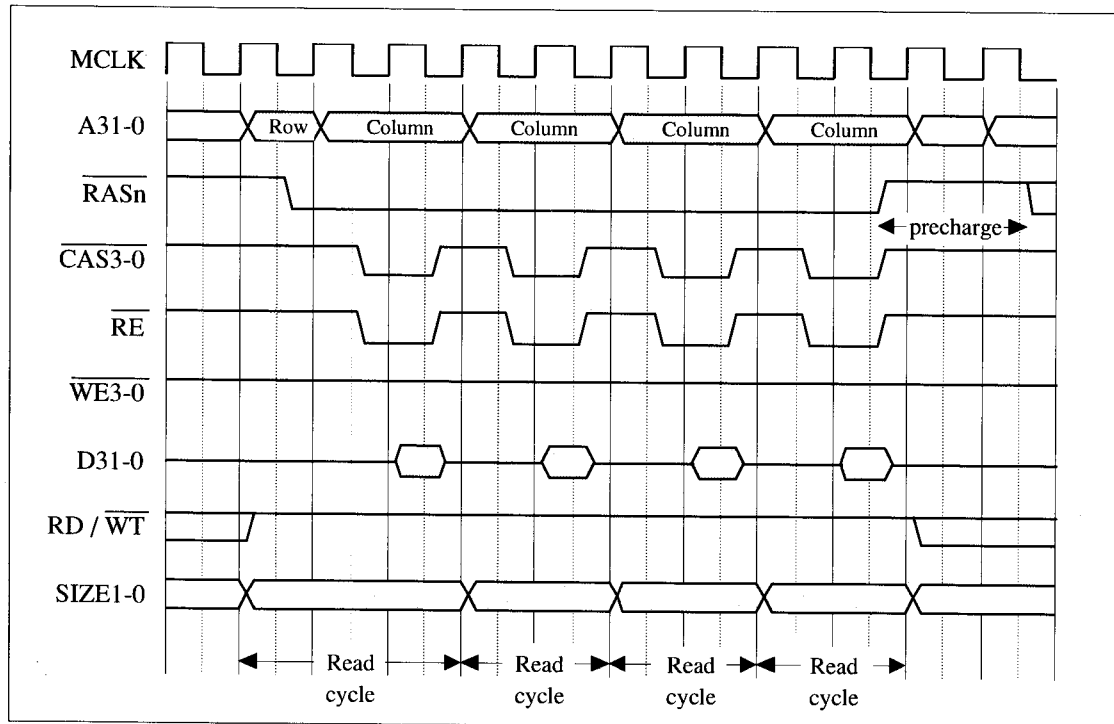
Fig. 5-8-20 32-bit DRAM Page Mode Read Timing (When FRQ = 0)
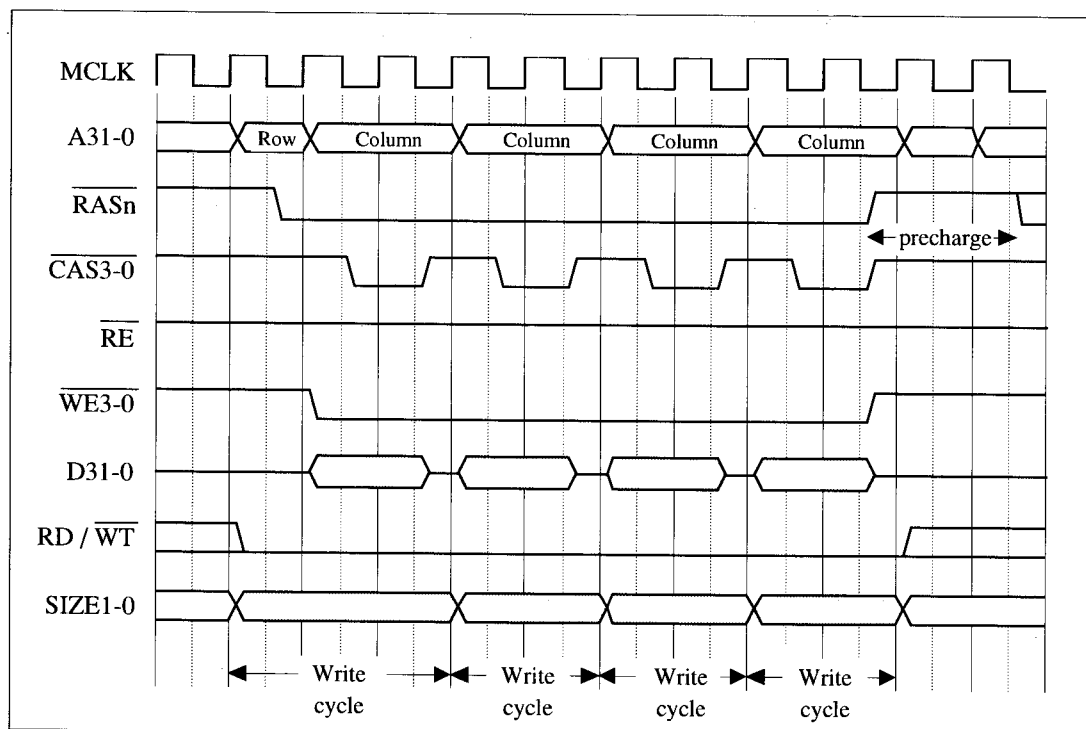
Fig. 5-8-21 32-bit DRAM Page Mode Write Timing (When FRQ = 0)
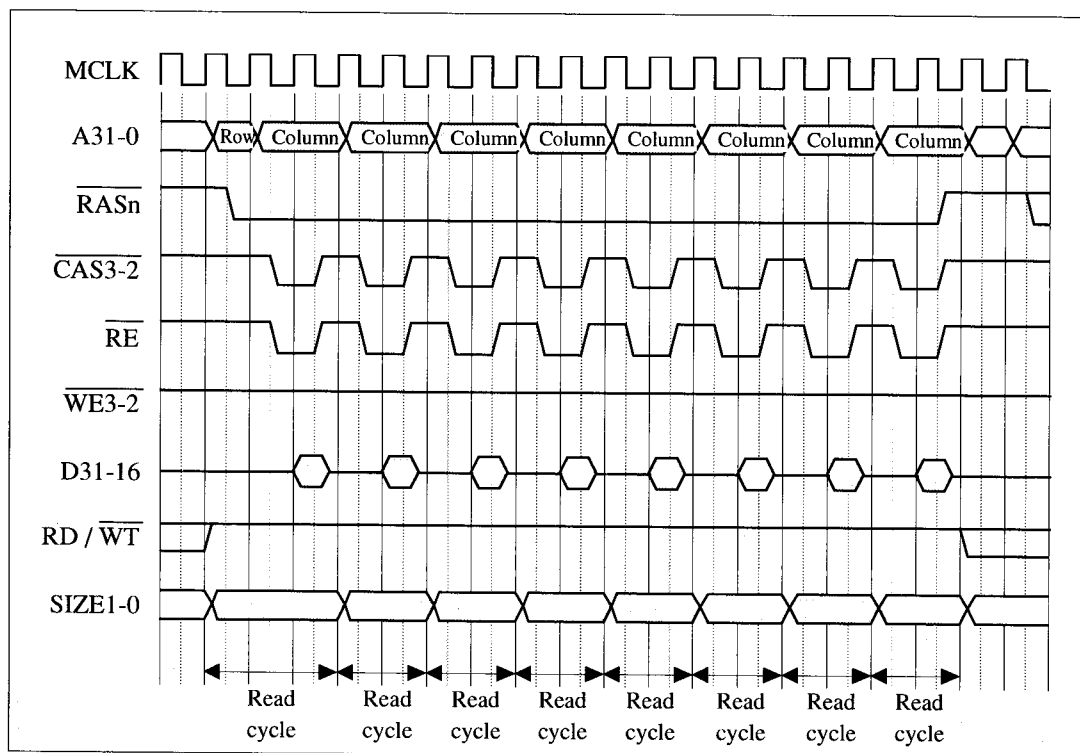
Fig. 5-8-22 16-bit DRAM Page Mode Read Timing (When FRQ = 0)
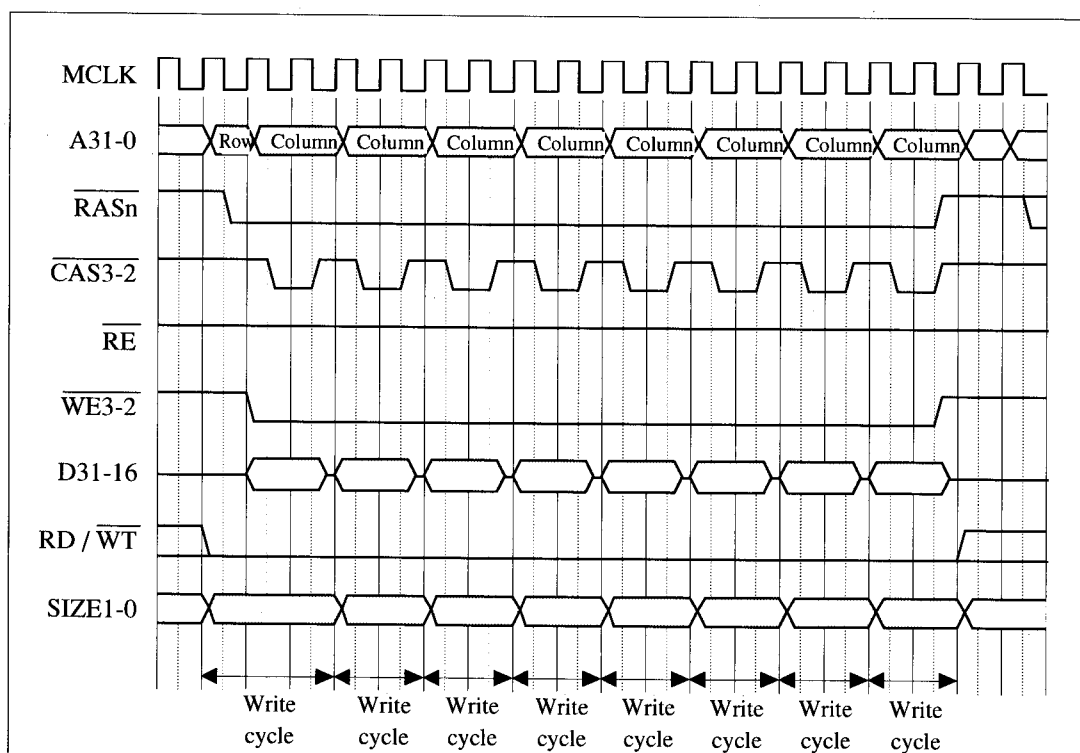
Fig. 5-8-23 16-bit DRAM Page Mode Write Timing (When FRQ = 0)

DRAM page mode (when FRQ = 1)

Just as when FRQ = 0, if the PAGE bit in the DRAM control register is set, page mode access is enabled, making high-speed access in page mode possible for consecutive accesses to DRAM.

The byte specification by WE3 through 0 is set by clearing the BnCAS bit in memory control registers 1 through 3. The RAS signal and, in the case of a write, the WE signal are negated as soon as the consecutive accesses to DRAM are completed.

The byte specification by CAS3 through 0 is set by setting the BnCAS bit in memory control registers 1 through 3. Just as in the case of byte specification by WE3 through WE0, the RAS signal and, in the case of a write, the WE signal are negated as soon as the consecutive accesses to DRAM are completed.

The address is shifted down and the row address is output according to the value of the SIZE1 and 0 bits in the DRAM control register. The RASn signal ("n" corresponds to the block number) and the CAS signal are then asserted according to the values of the RTC bit, RPCP bit and the WC3 through 0 bits in the DRAM control register.

Fig. 5-8-24 shows the page mode read timing with a 32-bit bus, and Fig. 5-8-26 shows the page mode read timing with a 16-bit bus. Fig. 5-8-25 shows the page mode write timing with a 32-bit bus, and Fig. 5-8-27 shows the page mode write timing with a 16-bit bus.
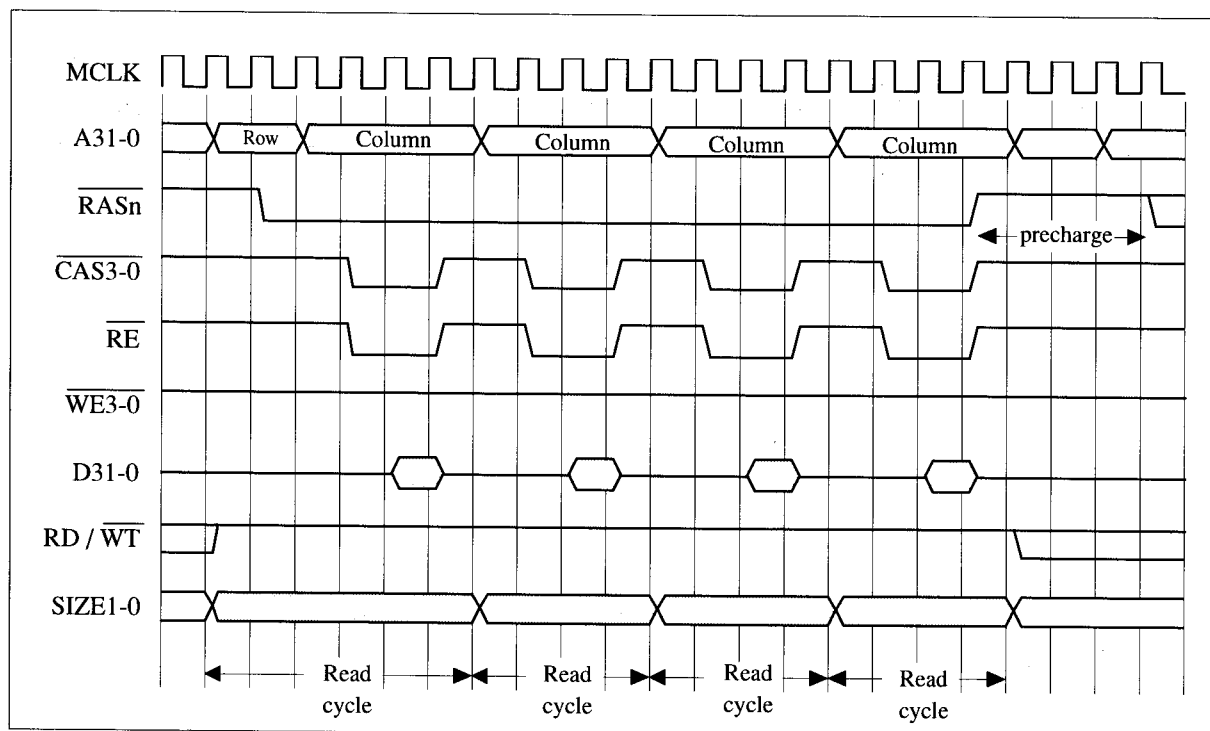
Fig. 5-8-24 32-bit DRAM Page Mode Read Timing (When FRQ = 1)
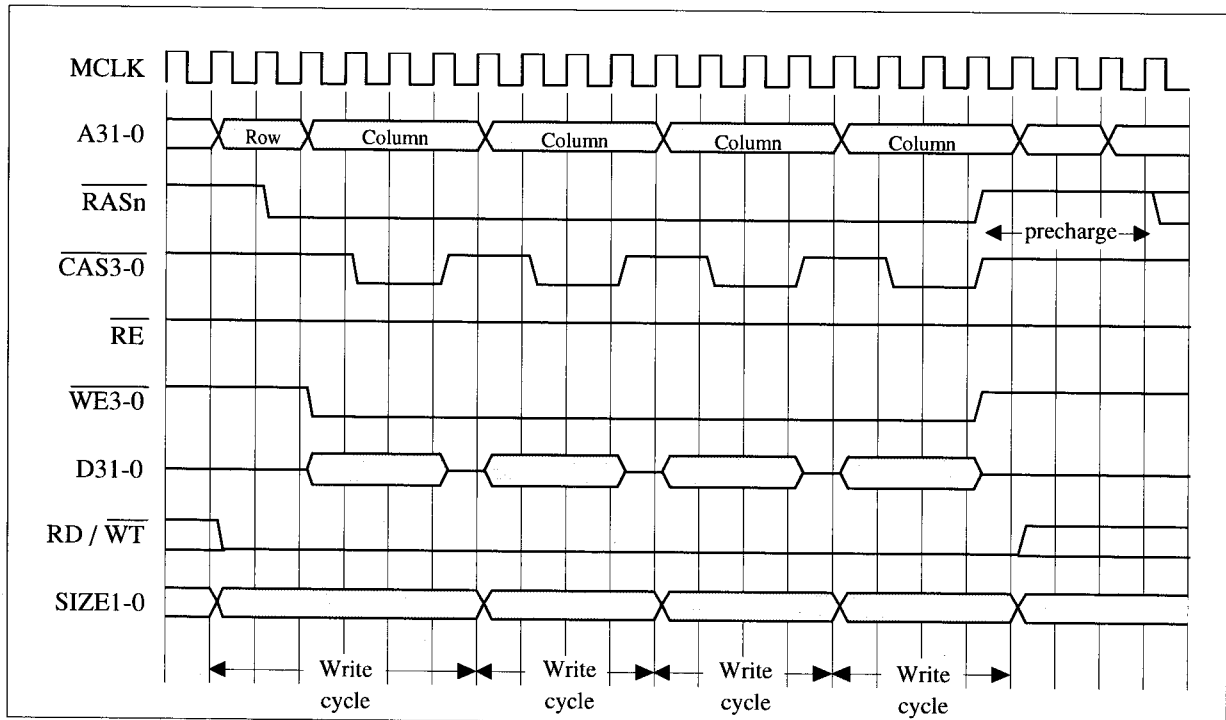
Fig. 5-8-25 32-bit DRAM Page Mode Write Timing (When FRQ = 1)
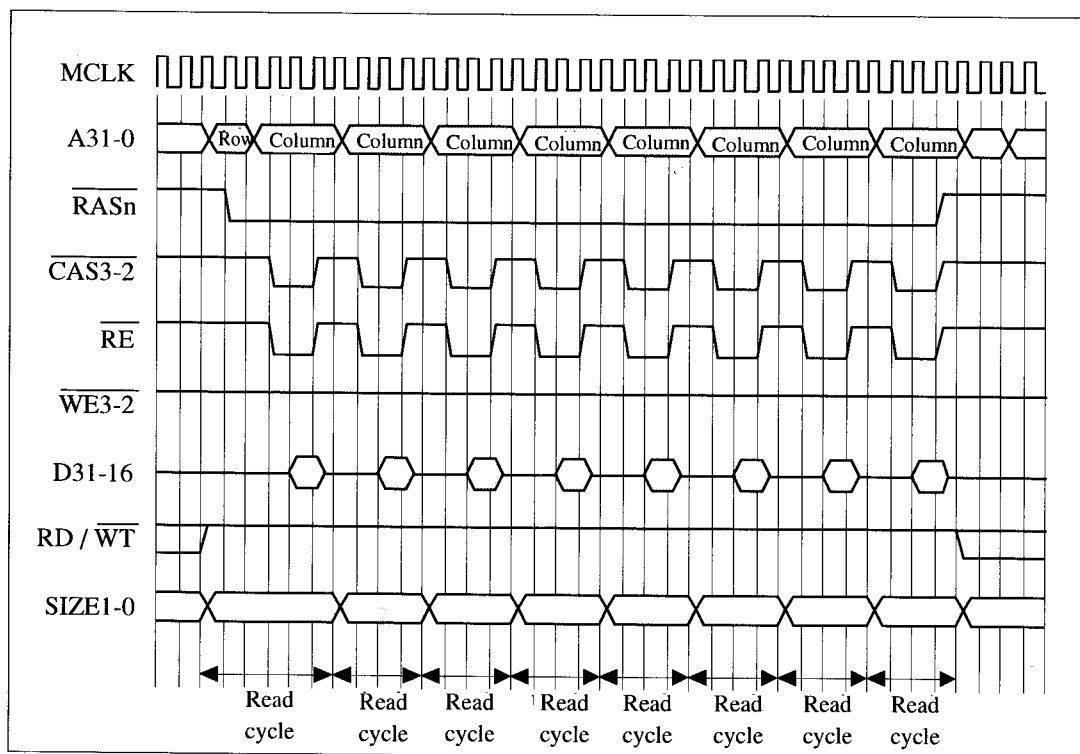
Fig. 5-8-26 16-bit DRAM Page Mode Read Timing (When FRQ = 1)
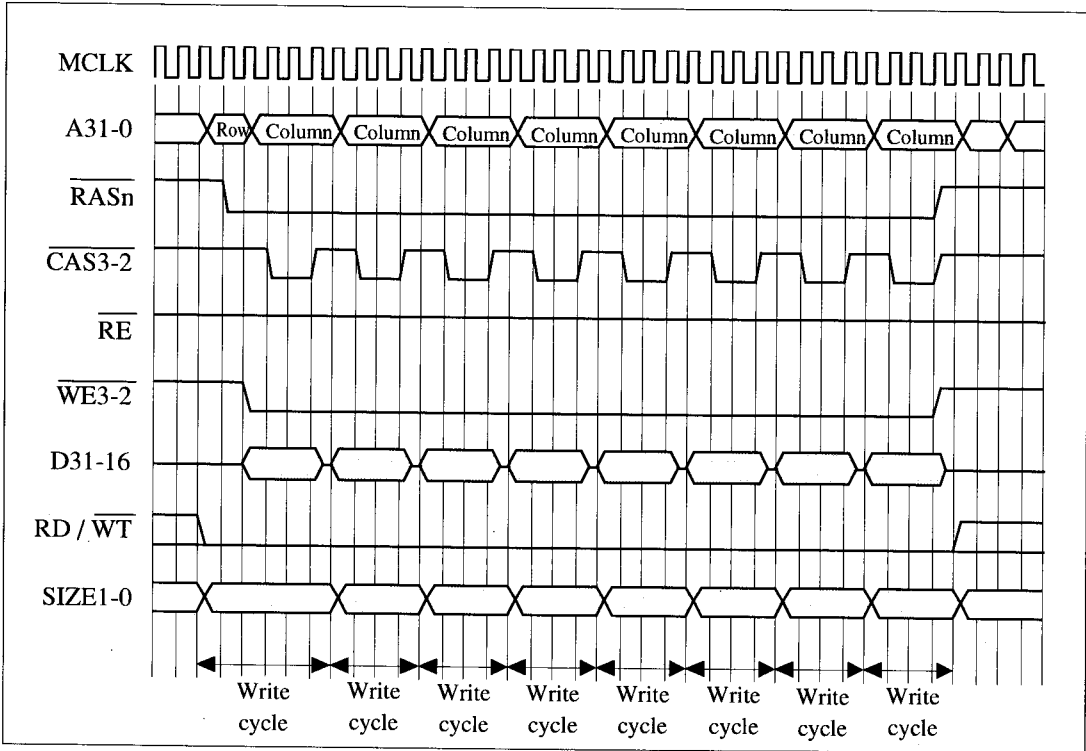
Fig. 5-8-27 16-bit DRAM Page Mode Write Timing (When FRQ = 1)

5.8.5.4 Refresh

If the REFE bit in the DRAM control register is set, CAS-before-RAS refresh is performed at the interval set by the refresh count register. Fig. 5-8-28 illustrates the refresh operation concept. The refresh interval is the product of the value of the REFC bits in the refresh count register and the SYSCLK frequency. If the REFE bit in the DRAM control register is set, the refresh count register operates as a down-counter that counts down from the REFC value to 0. The refresh operation is executed once in an idle external bus cycle during the period while the refresh count value is counted down from REFC to 0. If, due to a DMA access or other such operation, there is no idle external bus cycle before the refresh count value reaches 0, then a refresh cycle is inserted right after the bus cycle that is being executed at the moment the refresh count value reaches 0 is completed. (If a DMA burst transfer is in progress, the burst transfer is interrupted.)

Regarding the output timing of the RAS/CAS signals during the refresh operation, the RAS assertion period can be set through the RERP bit in the DRAM control register as shown in Fig. 5-8-29. The number of cycles is given in terms of the number of MCLK cycles.

In 32-bit bus mode, when all of the blocks that are set in blocks 1 through 4 of DRAM are set for WE byte specification, CAS3 is asserted; if CAS byte specification is set, CAS3 through 0 are asserted.

In 16-bit bus mode, when all of the blocks that are set in blocks 1 through 4 of DRAM are set for WE byte specification, CAS3 is asserted, just as in 32-bit bus mode; however, if CAS byte specification is set, only CAS3 and CAS2 are asserted.

Figs. 5-8-30 and 5-8-31 show the timing of a CAS-before-RAS refresh operation when the RERP bit in DRAMCTR is "0".
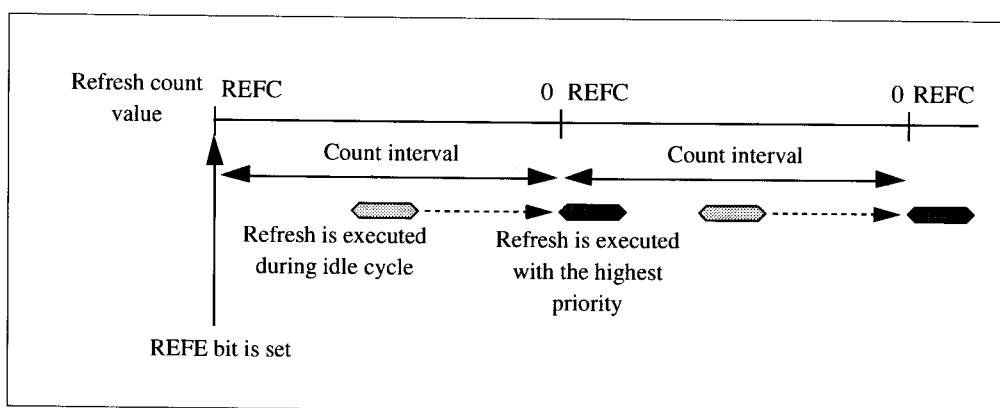
Fig. 5-8-28 DRAM Refresh Operation
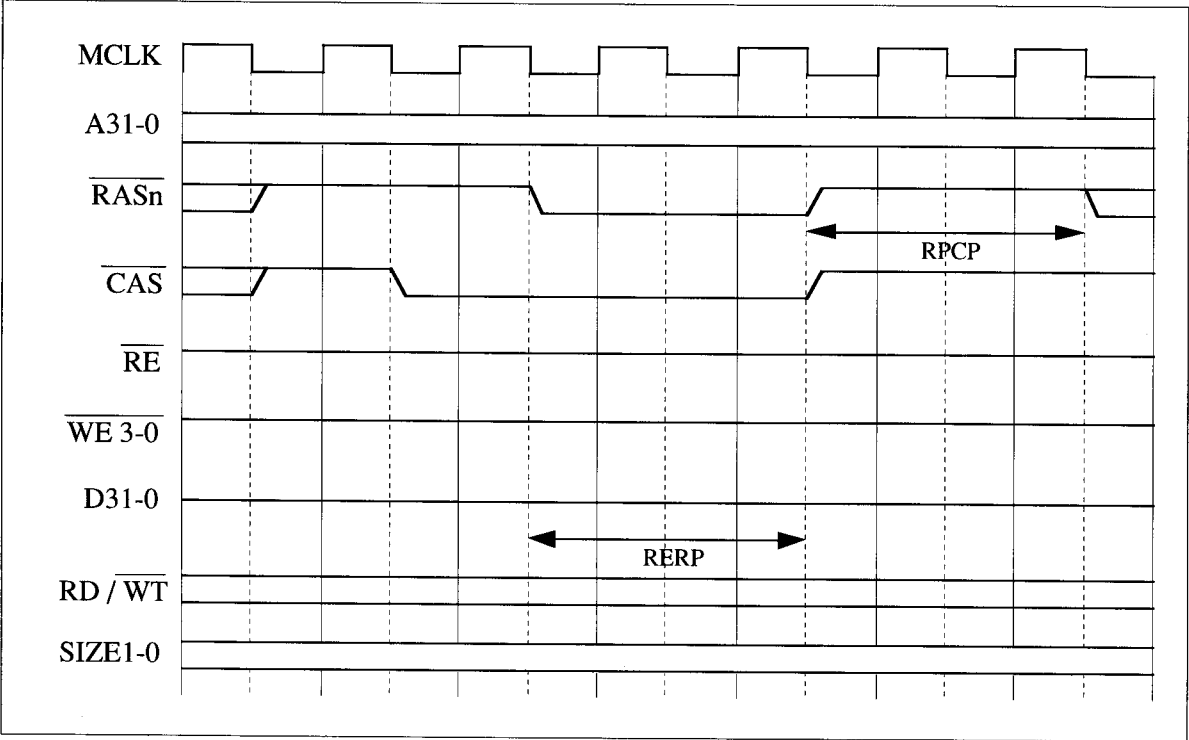
Fig. 5-8-29 Setting the DRAM Refresh Timing
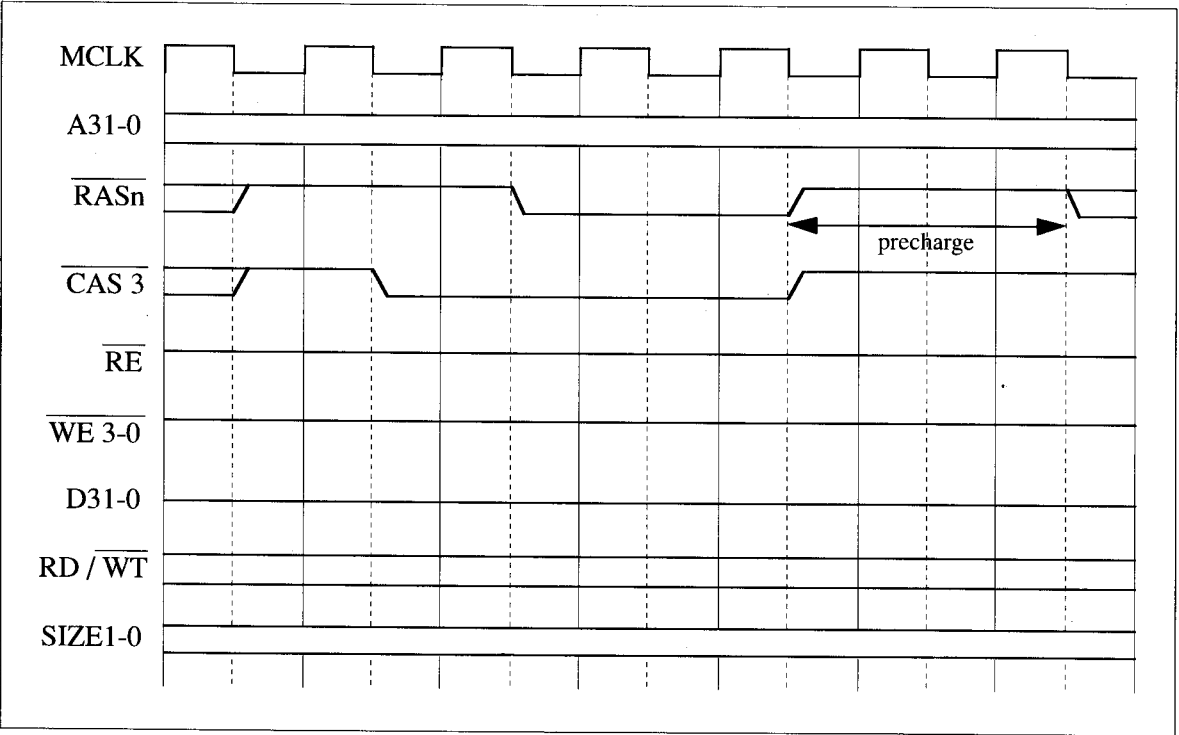
Fig. 5-8-30 DRAM Refresh Timing
(When all blocks that are set in blocks 1 through 4 of DRAM are set for WE byte specification, and FRQ = 0)
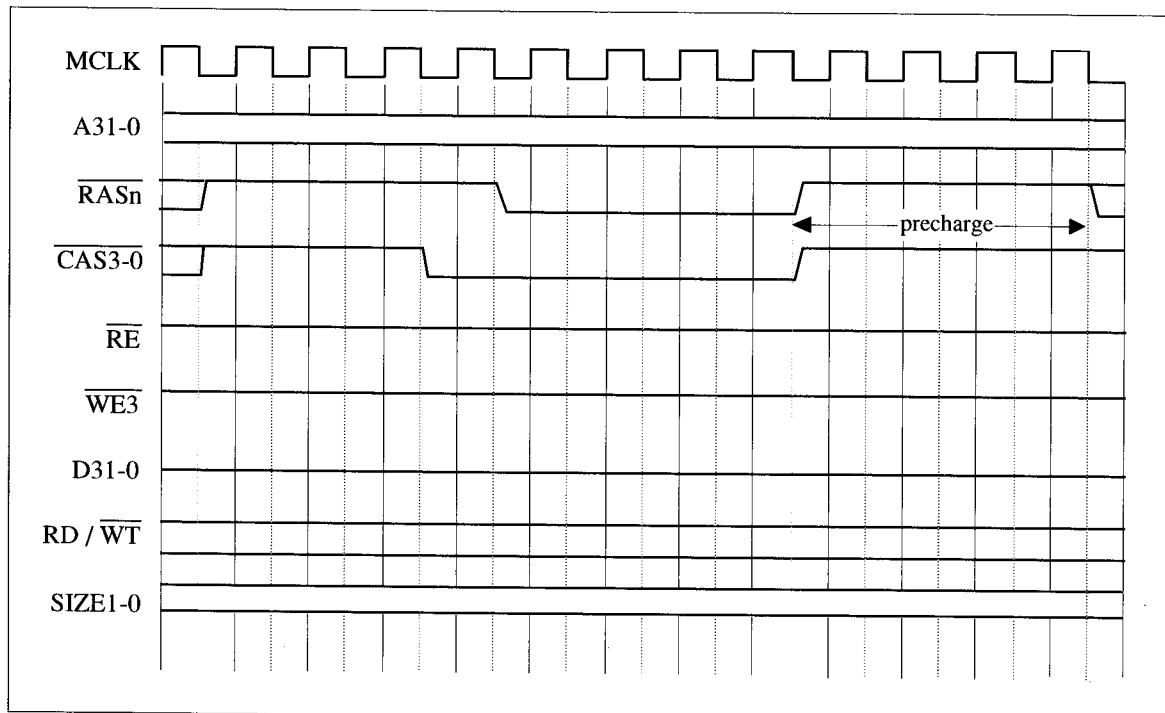
Fig. 5-8-31 DRAM Refresh Timing
(When CAS byte specification is set, and FRQ = 1)

5.8.5.5   Connection Examples

Fig. 5-8-32 shows an example of external memory connection when the DRAM byte specification is made through signals WE3 through 0, and Fig. 5-8-33 shows an example of external memory connection when the DRAM byte specification is made through signals CAS3 through 0. The memory configuration is listed below.

Block 0: 16-bit bus width, 4-Mbit ROM (262144 words × 16 bits)

Block 1: 32-bit bus width, 4-Mbit DRAM (262144 words × 16 bits) × 2

Block 2: 8-bit bus width, 1-Mbit SRAM (131072 words × 8 bits)
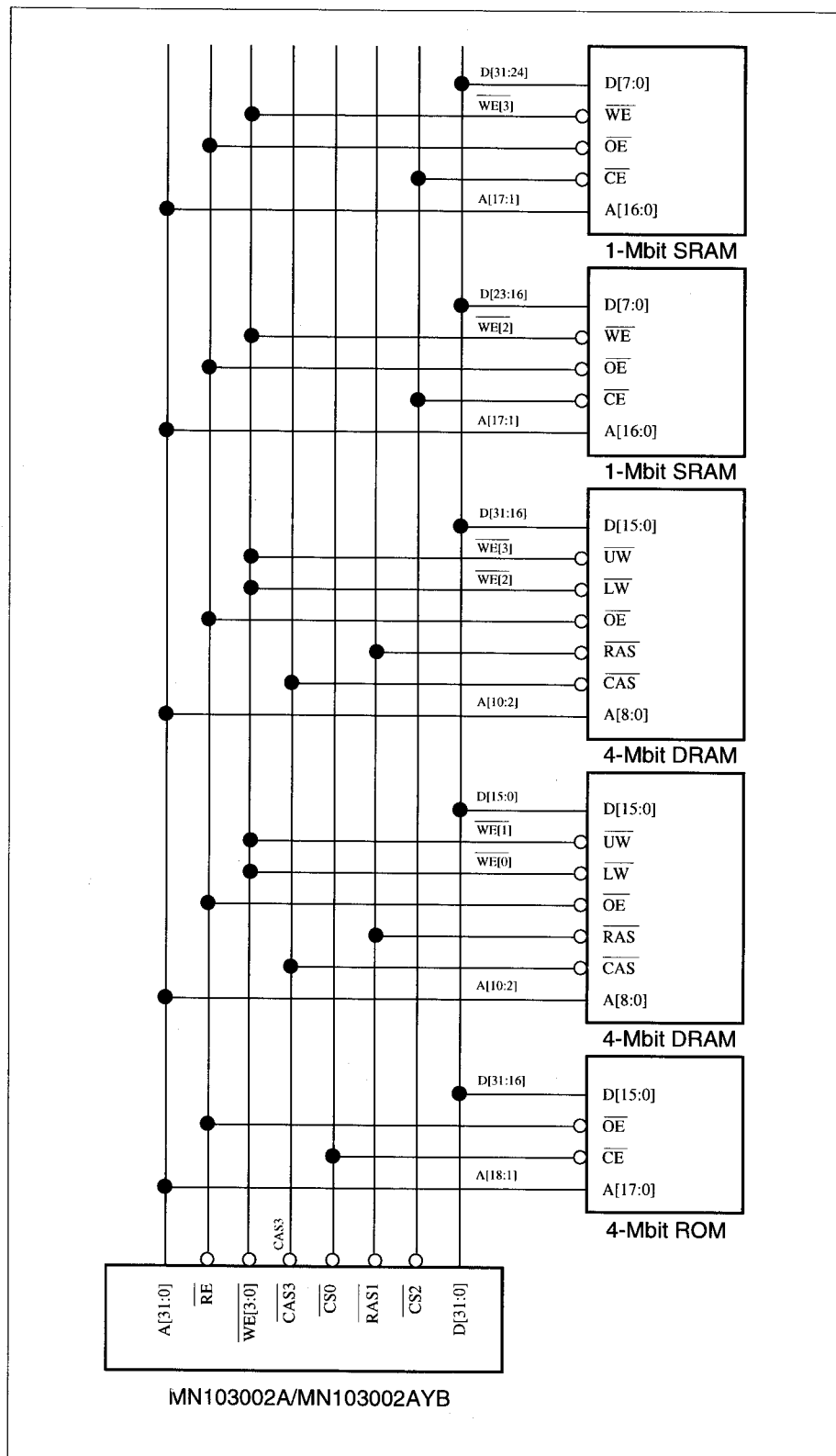
Fig. 5-8-32 Memory Connection Example (When DRAM Byte Specification Is By WE)
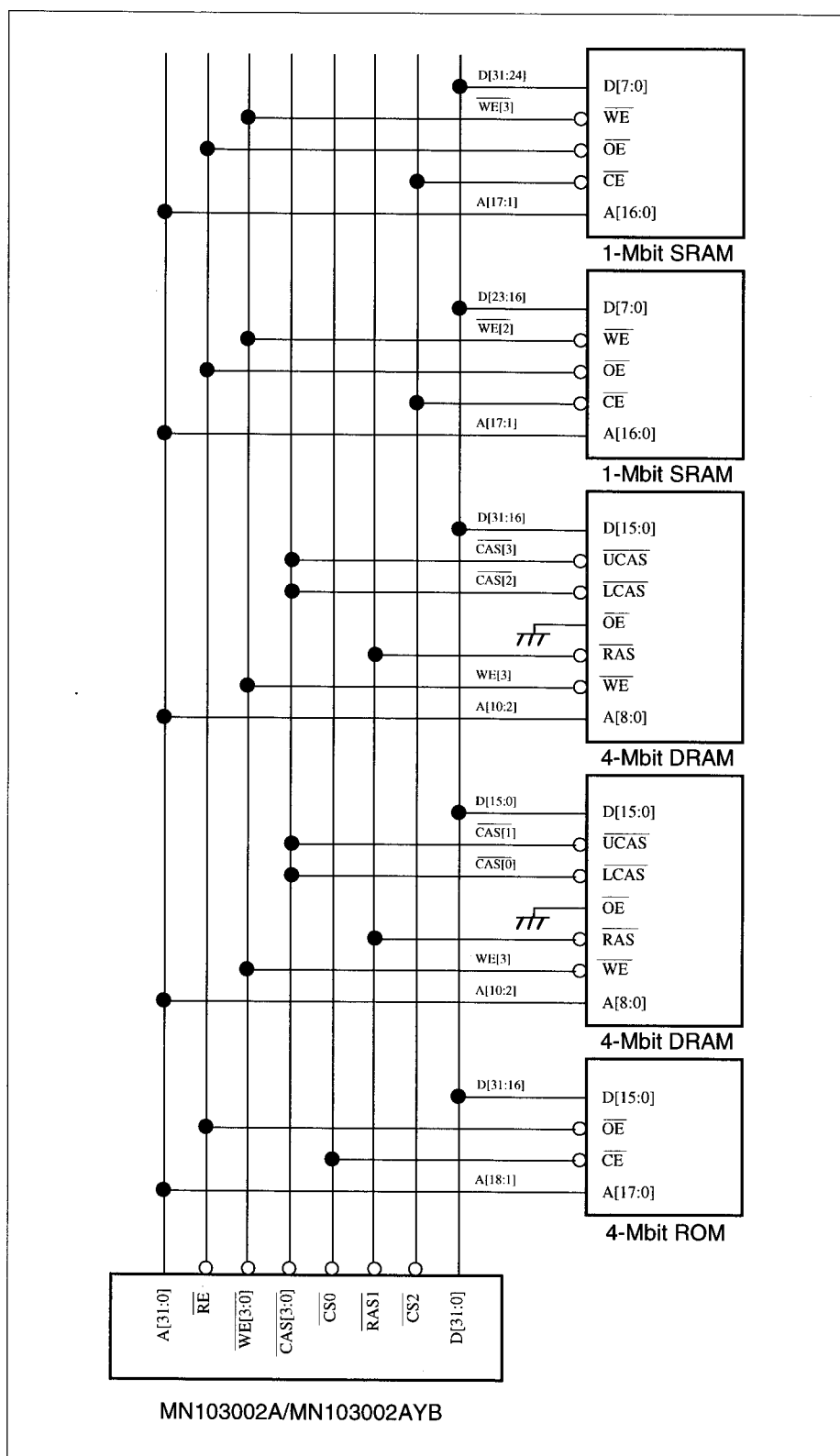
Fig. 5-8-33 Memory Connection Example (When DRAM Byte Specification Is By CAS)

5.8.6 Bus Arbitration

In the MN103002A/MN103002AYB, bus arbitration is implemented through the bus request signal (BR) and the bus grant signal (BG). If an external device asserts the BR signal, then once the current bus access being executed is completed, the BG signal is asserted and the bus authority is released to the external device. Once the BR signal is negated, this LSI negates the BG signal in order to re-acquire the bus authority. However, if a refresh request is generated by the DRAM control circuit within this LSI while the bus authority has been released to an external device, this LSI negates the BG signal and requests the bus authority back from the external device. The external device then negates the BR signal in response, and the refresh is executed. Note that bus arbitration is performed in synchronization with SYSCLK. Fig. 5-8-34 shows the timing for releasing the bus authority to an external device, and Fig. 5-8-35 shows the timing when a refresh request is generated while the bus authority has been released. A31 to 0, CSn, RE, WE3 to 0, RAS, and CAS3 to 0 are always output when the MN103002A/MN103002AYB has the bus authority (BG = 1), and go to high impedance when the bus authority is released (BG = 0).
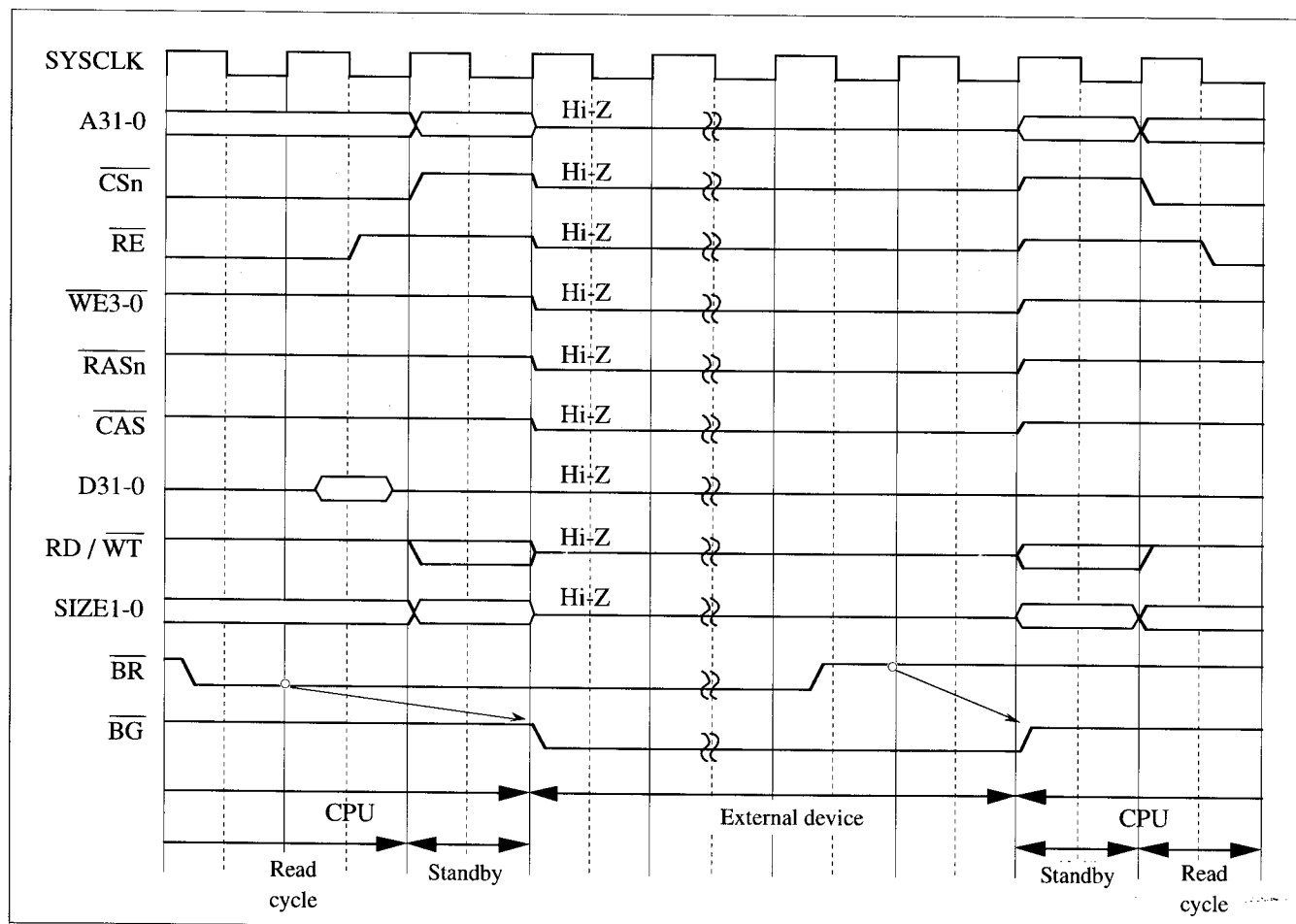
Fig. 5-8-34 Bus Arbitration Timing 1 (Bus Authority Release/Bus Authority Acquisition)
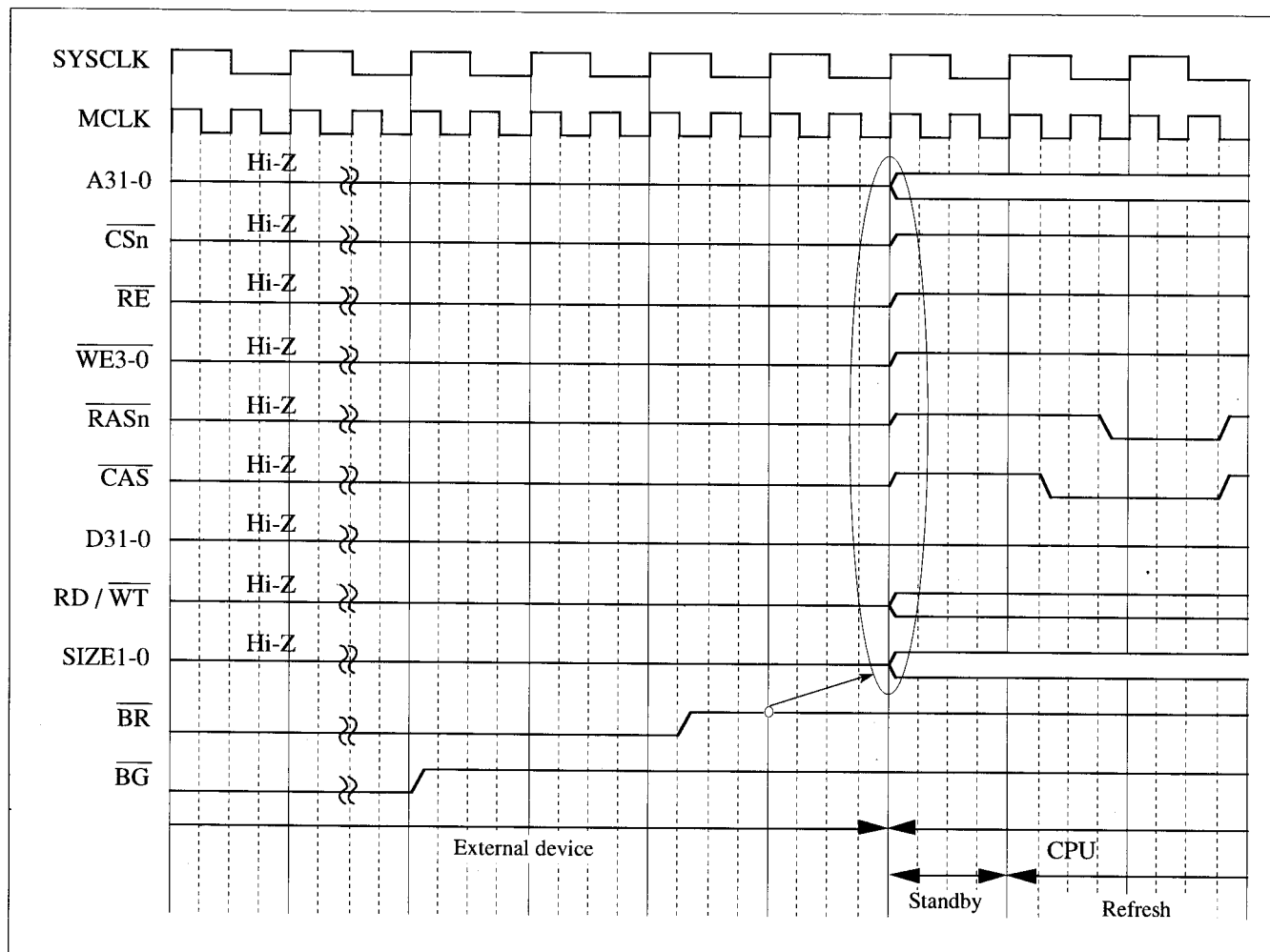
Fig. 5-8-35 Bus Arbitration Timing 2
(Refresh Request Generated While Bus Authority Has Been Released, and FRQ = 0)

5.9   Cautions

These cautions concern the BC. These cautions must be heeded, since failure to do so may result in misoperation.

1. Interrupts are prohibited and the bus is locked (occupied by the CPU) when BSET or BCLR is being executed. However, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request or DMA transfer may be interposed between the data read and data write by the BSET or BCLR instruction.

    If the atomic bus cycles of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken.

    1. A program in which a BSET or BCLR instruction is executed should be placed in instruction cache. Note that an external memory access is occurred, if a cache miss occurs in the instruction cache.

    2. Program so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

Chapter 6.   DMA Controller

6

6.1 Overview

The MN103002A/MN103002AYB has a built-in four-channel DMAC (Direct Memory Access Controller) that permits high-speed data transfers between external devices, internal I/O (except for the CPU's internal I/O registers) without using the CPU.

6.2 Features

The DMAC has the following features.
● Number of channels: 4 channels
● Transfer unit: 8-bit, 16-bit, or 32-bit units
● Maximum transfer count: 65536
● Startup sources:
- External requests: Transfer requests from the DMR3 to 0 pin2 (four channels)
- External interrupts: External interrupts from the IRQ1 pin and the IRQ0 pin
- Internal interrupts: TM 2, 3: underflow

    TM6A: compare/capture

    Serial 0, 1, 2: Transmission end/reception end
- Software execution: Set enable bit in the control register

● Transfer types
- Two bus cycle transfer

    Addressing is handled on both the source side and the destination side, and the transfer is accomplished over two bus cycles, a read cycle and a write cycle.
- One bus cycle transfer

    Transfers between external memory and an external device that supports the acknowledge function are performed in one bus cycle.
    • Transfer from an external device to external memory

    While the external device is accessed through the DMK signal and the data is read, the external memory is addressed, and then the data is written on the bus.
    • Transfer from external memory to an external device

    While external memory is addressed and the data is read, the external device is accessed through the DMK signal and the data is written on the bus.

● Addressing modes

    "Fixed," "increment" or "decrement" can be specified for the source address and the destination address, respectively. Incrementation and decrementation are performed automatically in accordance with the size of the transfer unit.

- ● Transfer modes
  - Single-word transfer

    In this mode, transfers are performed the required number of times, releasing the bus to the CPU after each word is transferred.
  - Burst transfer

    In this mode, transfers are performed consecutively the required number of times.
  - Intermittent transfer

    In this mode, transfers are performed continuously, releasing the bus to the CPU after a set number of transfers.

- ● Priority ranking

    The priority ranking of the channels is as follows: DMA ch0 > DMA ch3.

    If there are simultaneous transfer requests from multiple channels, the transfer on the channel with the highest priority ranking is executed. If, while a DMA transfer is in progress, there is a transfer request on a channel with a higher priority ranking, the transfer with the higher priority is executed after the next time the bus is released (after one word is transferred if single-word transfer mode is in effect, after the transfer is completed if burst transfer mode is in effect, or after the transfer for the set number of intermittent cycle transfers is completed if intermittent transfer mode is in effect).

- ● Transfer requests from external sources

    The DMR signal can be used to issue a transfer request from an external source. Edge detection is supported for the DMR signal sampling timing.

- ● DMA interruption

    When a non-maskable interrupt, external bus request, or DRAM refresh request is issued, the DMA transfer is interrupted. Transfer is resumed after the external bus has been used or the DRAM refresh operation has been completed.

- ● DMA forced end

    A DMA transfer can be forcibly ended by writing a "0" to the corresponding DMnTEN bit in the DMA control register. Be careful not to overwrite any other bits in the DMA control register.

6.3 Description of DMA Registers

The DMAC includes the registers that are listed in Table 6-3-1. These registers set the transfer initiation source, the transfer word count, etc.

Table 6-3-1 List of Registers

Address	Name	Symbol	Number of bits	Initial value	Address size
x'32000100	DMA0 control register	DM0CTR	32	x'00000000	8, 16, 32
x'32000104	DMA0 source address register	DM0SRC	32	x'XXXXXXXX	32
x'32000108	DMA0 destination address register	DM0DST	32	x'XXXXXXXX	32
x'3200010C	DMA0 transfer word count register	DM0CNT	16	x'0000	16
x'3200010E	DMA0 intermittent cycle register	DM0CYC	8	x'00	8
x'32000200	DMA1 control register	DM1CTR	32	x'00000000	8, 16, 32
x'32000204	DMA1 source address register	DM1SRC	32	x'XXXXXXXX	32
x'32000208	DMA1 destination address register	DM1DST	32	x'XXXXXXXX	32
x'3200020C	DMA1 transfer word count register	DM1CNT	16	x'0000	16
x'3200020E	DMA1 intermittent cycle register	DM1CYC	8	x'00	8
x'32000400	DMA2 control register	DM2CTR	32	x'00000000	8, 16, 32
x'32000404	DMA2 source address register	DM2SRC	32	x'XXXXXXXX	32
x'32000408	DMA2 destination address register	DM2DST	32	x'XXXXXXXX	32
x'3200040C	DMA2 transfer word count register	DM2CNT	16	x'0000	16
x'3200040E	DMA2 intermittent cycle register	DM2CYC	8	x'00	8
x'32000800	DMA3 control register	DM3CTR	32	x'00000000	8, 16, 32
x'32000804	DMA3 source address register	DM3SRC	32	x'XXXXXXXX	32
x'32000808	DMA3 destination address register	DM3DST	32	x'XXXXXXXX	32
x'3200080C	DMA3 transfer word count register	DM3CNT	16	x'0000	16
x'3200080E	DMA3 intermittent cycle register	DM3CYC	8	x'00	8

6.3.1 DMA Control Registers

DMA control register 0

Register symbol: DM0CTR

Address: x'32000100

Purpose: Sets the transfer parameters for DMA channel 0.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	-	-	-	-	-	-	-	DM0 RQF	-	-	-	-	-	-	-	DM0 TEN
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DM0 RQM	DM0 UT1	DM0 UT0	DM0 TM1	DM0 TM0	DM0 DIR	DM0 DAM1	DM0 DAM0	DM0 SAM1	DM0 SAM0	DM0 ST	DM0 BG4	DM0 BG3	DM0 BG2	DM0 BG1	DM0 BG0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0	DM0BG0	DMA channel 0 initiation source (LSB)
1	DM0BG1	DMA channel 0 initiation source
2	DM0BG2	DMA channel 0 initiation source
3	DM0BG3	DMA channel 0 initiation source
4	DM0BG4	DMA channel 0 initiation source (MSB)

00000: Software source

00001: Setting prohibited

00010: Serial 0 transmission end source

00011: Serial 0 reception end source

00100: Serial 1 transmission end source

00101: Serial 1 reception end source

00110: Serial 2 transmission end source

00111: Serial 2 reception end source

01000: Timer 2 underflow source

01001: Timer 3 underflow source

01010: Timer 6A compare/capture source

01011: Setting prohibited

01100: IRQ0 input source

01101: IRQ1 input source

01110: External request 0 source

01111: External request 1 source

10000 to 11111: Setting prohibited

<Continued>

<Continued>

Bit position	Bit name	Description
5	DM0ST	DMA channel 0 transfer type

0: Two bus cycle transfer

1: One bus cycle transfer

(Transfer between external memory and an external device that supports the acknowledge function)

| 6 | DM0SAM0 | DMA channel 0 transfer addressing mode on source side (LSB) |
| 7 | DM0SAM1 | DMA channel 0 transfer addressing mode on source side (MSB) |

00: Increment   01: Decrement

10: Fixed   11: Setting prohibited

| 8 | DM0DAM0 | DMA channel 0 transfer addressing mode on destination side (LSB) |
| 9 | DM0DAM1 | DMA channel 0 transfer addressing mode on destination side (MSB) |

00: Increment   01: Decrement

10: Fixed   11: Setting prohibited

| 10 | DM0DIR | DMA channel 0 transfer direction |

(Direction of external device that supports the acknowledge function in a one bus cycle transfer)

0: Source   1: Destination

| 11 | DM0TM0 | DMA channel 0 transfer mode (LSB) |
| 12 | DM0TM1 | DMA channel 0 transfer mode (MSB) |

00: Burst transfer   01: Single-word transfer

10: Intermittent transfer   11: Setting prohibited

| 13 | DM0UT0 | DMA channel 0 transfer unit (LSB) |
| 14 | DM0UT1 | DMA channel 0 transfer unit (MSB) |

00: 8 bits   01: 16 bits

10: 32 bits   11: Setting prohibited

| 15 | DM0RQM | DMA channel 0 DMR signal mode |

Always set to "1".

| 16 | DM0TEN | DMA channel 0 DMA transfer enable |

0: DMA transfer disabled   1: DMA transfer enabled

| 24 | DM0RQF | DMA channel 0 DMA transfer request flag |

(Automatically set when a startup source is generated; automatically reset after one transfer operation in single-word transfer mode, after all transfers have been completed in intermittent transfer or burst transfer mode, or in the event of a forced termination)

0: No request   1: Request

> **!** Note that the external request sources differ from those in DMA control registers 2 and 3.

DMA control register 1

Register symbol: DM1CTR

Address: x'32000200

Purpose: Sets the transfer parameters for DMA channel 1.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	-	-	-	-	-	-	-	DM1 RQF	-	-	-	-	-	-	-	DM1 TEN
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DM1 RQM	DM1 UT1	DM1 UT0	DM1 TM1	DM1 TM0	DM1 DIR	DM1 DAM1	DM1 DAM0	DM1 SAM1	DM1 SAM0	DM1 ST	DM1 BG4	DM1 BG3	DM1 BG2	DM1 BG1	DM1 BG0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0	DM1BG0	DMA channel 1 initiation source (LSB)
1	DM1BG1	DMA channel 1 initiation source
2	DM1BG2	DMA channel 1 initiation source
3	DM1BG3	DMA channel 1 initiation source
4	DM1BG4	DMA channel 1 initiation source (MSB)

00000: Software source

00001: Setting prohibited

00010: Serial 0 transmission end source

00011: Serial 0 reception end source

00100: Serial 1 transmission end source

00101: Serial 1 reception end source

00110: Serial 2 transmission end source

00111: Serial 2 reception end source

01000: Timer 2 underflow source

01001: Timer 3 underflow source

01010: Timer 6A compare/capture source

01011: Setting prohibited

01100: IRQ0 input source

01101: IRQ1 input source

01110: External request 0 source

01111: External request 1 source

10000 to 11111: Setting prohibited

<Continued>

\<Continued\>

Bit position	Bit name	Description
5	DM1ST	DMA channel 1 transfer type 0: Two bus cycle transfer 1: One bus cycle transfer (Transfer between external memory and an external device that supports the acknowledge function)
6	DM1SAM0	DMA channel 1 transfer addressing mode on source side (LSB)
7	DM1SAM1	DMA channel 1 transfer addressing mode on source side (MSB) 00: Increment 01: Decrement 10: Fixed 11: Setting prohibited
8	DM1DAM0	DMA channel 1 transfer addressing mode on destination side (LSB)
9	DM1DAM1	DMA channel 1 transfer addressing mode on destination side (MSB) 00: Increment 01: Decrement 10: Fixed 11: Setting prohibited
10	DM1DIR	DMA channel 1 transfer direction (Direction of external device that supports the acknowledge function in a one bus cycle transfer) 0: Source 1: Destination
11	DM1TM0	DMA channel 1 transfer mode (LSB)
12	DM1TM1	DMA channel 1 transfer mode (MSB) 00: Burst transfer 01: Single-word transfer 10: Intermittent transfer 11: Setting prohibited
13	DM1UT0	DMA channel 1 transfer unit (LSB)
14	DM1UT1	DMA channel 1 transfer unit (MSB) 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
15	DM1RQM	DMA channel 1 DMR signal mode Always set to "1".
16	DM1TEN	DMA channel 1 DMA transfer enable 0: DMA transfer disabled 1: DMA transfer enabled
24	DM1RQF	DMA channel 1 DMA transfer request flag (Automatically set when a startup source is generated; automatically reset after one transfer operation in single-word transfer mode, after all transfers have been completed in intermittent transfer or burst transfer mode, or in the event of a forced termination) 0: No request 1: Request

Note that the external request sources differ from those in DMA control registers 2 and 3.

DMA control register 2

Register symbol: DM2CTR

Address: x'32000400

Purpose: Sets the transfer parameters for DMA channel 2.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	-	-	-	-	-	-	-	DM2 RQF	-	-	-	-	-	-	-	DM2 TEN
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DM2 RQM	DM2 UT1	DM2 UT0	DM2 TM1	DM2 TM0	DM2 DIR	DM2 DAM1	DM2 DAM0	DM2 SAM1	DM2 SAM0	DM2 ST	DM2 BG4	DM2 BG3	DM2 BG2	DM2 BG1	DM2 BG0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0	DM2BG0	DMA channel 2 initiation source (LSB)
1	DM2BG1	DMA channel 2 initiation source
2	DM2BG2	DMA channel 2 initiation source
3	DM2BG3	DMA channel 2 initiation source
4	DM2BG4	DMA channel 2 initiation source (MSB)

00000: Software source

00001: Setting prohibited

00010: Serial 0 transmission end source

00011: Serial 0 reception end source

00100: Serial 1 transmission end source

00101: Serial 1 reception end source

00110: Serial 2 transmission end source

00111: Serial 2 reception end source

01000: Timer 2 underflow source

01001: Timer 3 underflow source

01010: Timer 6A compare/capture source

01011: Setting prohibited

01100: IRQ0 input source

01101: IRQ1 input source

01110: External request 2 source

01111: External request 3 source

10000 to 11111: Setting prohibited

<Continued>

<Continued>

Bit position	Bit name	Description
5	DM2ST	DMA channel 2 transfer type 0: Two bus cycle transfer 1: One bus cycle transfer (Transfer between external memory and an external device that sup ports the acknowledge function)
6	DM2SAM0	DMA channel 2 transfer addressing mode on source side (LSB)
7	DM2SAM1	DMA channel 2 transfer addressing mode on source side (MSB) 00: Increment 01: Decrement 10: Fixed 11: Setting prohibited
8	DM2DAM0	DMA channel 2 transfer addressing mode on destination side (LSB)
9	DM2DAM1	DMA channel 2 transfer addressing mode on destination side (MSB) 00: Increment 01: Decrement 10: Fixed 11: Setting prohibited
10	DM2DIR	DMA channel 2 transfer direction (Direction of external device that supports the acknowledge function in a one bus cycle transfer) 0: Source 1: Destination
11	DM2TM0	DMA channel 2 transfer mode (LSB)
12	DM2TM1	DMA channel 2 transfer mode (MSB) 00: Burst transfer 01: Single-word transfer 10: Intermittent transfer 11: Setting prohibited
13	DM2UT0	DMA channel 2 transfer unit (LSB)
14	DM2UT1	DMA channel 2 transfer unit (MSB) 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
15	DM2RQM	DMA channel 2 DMR signal mode Always set to "1".
16	DM2TEN	DMA channel 2 DMA transfer enable 0: DMA transfer disabled 1: DMA transfer enabled
24	DM2RQF	DMA channel 2 DMA transfer request flag (Automatically set when a startup source is generated; automatically reset after one transfer operation in single-word transfer mode, after all transfers have been completed in intermittent transfer or burst transfer mode, or in the event of a forced termination) 0: No request 1: Request

Note that the external request sources differ from those in DMA control registers 0 and 1.

DMA control register 3

Register symbol: DM3CTR

Address: x'32000800

Purpose: Sets the transfer parameters for DMA channel 3.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	-	-	-	-	-	-	-	DM3 RQF	-	-	-	-	-	-	-	DM3 TEN
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DM3 RQM	DM3 UT1	DM3 UT0	DM3 TM1	DM3 TM0	DM3 DIR	DM3 DAM1	DM3 DAM0	DM3 SAM1	DM3 SAM0	DM3 ST	DM3 BG4	DM3 BG3	DM3 BG2	DM3 BG1	DM3 BG0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0	DM2BG0	DMA channel 3 initiation source (LSB)
1	DM2BG1	DMA channel 3 initiation source
2	DM2BG2	DMA channel 3 initiation source
3	DM2BG3	DMA channel 3 initiation source
4	DM2BG4	DMA channel 3 initiation source (MSB)

00000: Software source

00001: Setting prohibited

00010: Serial 0 transmission end source

00011: Serial 0 reception end source

00100: Serial 1 transmission end source

00101: Serial 1 reception end source

00110: Serial 2 transmission end source

00111: Serial 2 reception end source

01000: Timer 2 underflow source

01001: Timer 3 underflow source

01010: Timer 6A compare/capture source

01011: Setting prohibited

01100: IRQ0 input source

01101: IRQ1 input source

01110: External request 2 source

01111: External request 3 source

10000 to 11111: Setting prohibited

<Continued>

<Continued>

Bit position	Bit name	Description
5	DM3ST	DMA channel 3 transfer type

0: Two bus cycle transfer

1: One bus cycle transfer

(Transfer between external memory and an external device that sup

ports the acknowledge function)

| 6 | DM3SAM0 | DMA channel 3 transfer addressing mode on source side (LSB) |
| 7 | DM3SAM1 | DMA channel 3 transfer addressing mode on source side (MSB) |

00: Increment   01: Decrement

10: Fixed   11: Setting prohibited

| 8 | DM3DAM0 | DMA channel 3 transfer addressing mode on destination side (LSB) |
| 9 | DM3DAM1 | DMA channel 3 transfer addressing mode on destination side (MSB) |

00: Increment   01: Decrement

10: Fixed   11: Setting prohibited

| 10 | DM3DIR | DMA channel 3 transfer direction |

(Direction of external device that supports the acknowledge function in

a one bus cycle transfer)

0: Source   1: Destination

| 11 | DM3TM0 | DMA channel 3 transfer mode (LSB) |
| 12 | DM3TM1 | DMA channel 3 transfer mode (MSB) |

00: Burst transfer   01: Single-word transfer

10: Intermittent transfer   11: Setting prohibited

| 13 | DM3UT0 | DMA channel 3 transfer unit (LSB) |
| 14 | DM3UT1 | DMA channel 3 transfer unit (MSB) |

00: 8 bits   01: 16 bits

10: 32 bits   11: Setting prohibited

| 15 | DM3RQM | DMA channel 3 DMR signal mode |

Always set to "1".

| 16 | DM3TEN | DMA channel 3 DMA transfer enable |

0: DMA transfer disabled   1: DMA transfer enabled

| 24 | DM3RQF | DMA channel 3 DMA transfer request flag |

(Automatically set when a startup source is generated; automatically reset

after one transfer operation in single-word transfer mode, after all transfers

have been completed in intermittent transfer or burst transfer mode, or in

the event of a forced termination)

0: No request   1: Request

**Note that the external request sources differ from those in DMA
control registers 0 and 1.**

6.3.2 DMA Source Address Registers

Register symbol: DMnSRC (n=0, 1, 2, 3)

Address: x'32000104 (n=0), x'32000204 (n=1),

x'32000404 (n=2), x'32000804 (n=3)

Purpose: These registers set the transfer source address for DMA channels 0 to 3.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	DMn SA31	DMn SA30	DMn SA29	DMn SA28	DMn SA27	DMn SA26	DMn SA25	DMn SA24	DMn SA23	DMn SA22	DMn SA21	DMn SA20	DMn SA19	DMn SA18	DMn SA17	DMn SA16
When reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DMn SA15	DMn SA14	DMn SA13	DMn SA12	DMn SA11	DMn SA10	DMn SA9	DMn SA8	DMn SA7	DMn SA6	DMn SA5	DMn SA4	DMn SA3	DMn SA2	DMn SA1	DMn SA0
When reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

During a DMA transfer, this register shows the next transfer source address. When the transfer source is an external device that supports the acknowledge function and one bus cycle transfer mode is in effect, the contents of DMnSRC are ignored. The transfer address that is set should conform with a 1-byte boundary (8 bits), 2-byte boundary (16 bits) or 4-byte boundary (32 bits), depending on the transfer byte unit.

6.3.3 DMA Destination Address Registers

Register symbol: DMnDST (n=0, 1, 2, 3)

Address: x'32000108 (n=0), x'32000208 (n=1),

x'32000408 (n=2), x'32000808 (n=3)

Purpose: These registers set the transfer destination address for DMA channels 0 to 3.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	DMn DA31	DMn DA30	DMn DA29	DMn DA28	DMn DA27	DMn DA26	DMn DA25	DMn DA24	DMn DA23	DMn DA22	DMn DA21	DMn DA20	DMn DA19	DMn DA18	DMn DA17	DMn DA16
When reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DMn DA15	DMn DA14	DMn DA13	DMn DA12	DMn DA11	DMn DA10	DMn DA9	DMn DA8	DMn DA7	DMn DA6	DMn DA5	DMn DA4	DMn DA3	DMn DA2	DMn DA1	DMn DA0
When reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

During a DMA transfer, this register shows the next transfer destination address. When the transfer destination is an external device that supports the acknowledge function and one bus cycle transfer mode is in effect, the contents of DMnDST are ignored. The transfer address that is set should conform with a 1-byte boundary (8 bits), 2-byte boundary (16 bits) or 4-byte boundary (32 bits), depending on the transfer byte unit.

6.3.4 DMA Transfer Word Count Registers

Register symbol: DMnCNT (n=0, 1, 2, 3)

Address: x'3200010C (n=0), x'3200020C (n=1),

x'3200040C (n=2), x'3200080C (n=3)

Purpose: These registers set the transfer word count for DMA channels 0 to 3.

Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	DMn CT15	DMn CT14	DMn CT13	DMn CT12	DMn CT11	DMn CT10	DMn CT9	DMn CT8	DMn CT7	DMn CT6	DMn CT5	DMn CT4	DMn CT3	DMn CT2	DMn CT1	DMn CT0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0 to 15	DMnCT	DMA transfer word count

x'0000 : 1 transfer x'FFFF : 65536 transfers

(maximum transfer count)

6.3.5 DMA Intermittent Cycle Registers

Register symbol: DMnCYC (n=0, 1, 2, 3)

Address: x'3200010E (n=0), x'3200020E (n=1),

x'3200040E (n=2), x'3200080E (n=3)

Purpose: These registers set the intermittent transfer cycle count for DMA channels 0 to 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	DMn CYC7	DMn CYC6	DMn CYC5	DMn CYC4	DMn CYC3	DMn CYC2	DMn CYC1	DMn CYC0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit position	Bit name	Description
0 to 7	DMnCYC	DMA intermittent cycle count

x'00 : 1 intermittent transfer

(Minimum number of intermittent cycles: releases bus to CPU after one DMA transfer)

x'FF : 256 intermittent transfers

(Maximum number of intermittent cycles: releases bus to CPU after 256 DMA transfers)

6.4 Description of Operation

6.4.1 Transfer Types

Table 6-4-1 lists the types of DMA transfers that can be supported. Note that in a one bus cycle transfer, the transfer source and the transfer destination are accessed in one bus cycle, while in a two bus cycle transfer, the transfer source and the transfer destination are accessed in two bus cycles. External devices that support the acknowledge function use one cycle transfer. The selection of one bus cycle transfer or two bus cycle transfer is made through the DMnST bits in the DMA control register.

Table 6-4-1 Devices to Which Transfer Is Possible

Transfer source/ transfer destination	Internal I/O [*1]	External memory	External device with acknowledge function
Internal I/O [*1]	2 bus cycles	2 bus cycles	Not possible
External memory	2 bus cycles	2 bus cycles	1 bus cycle
External device with acknowledge function	Not possible	1 bus cycles	Not possible

[*1] Excludes the I/O registers in the CPU and the I/O registers in the bus controller (the memory control registers, DMA registers, etc.). These registers cannot be the target of a DMA transfer.

6.4.2. Transfer Mode

The transfer modes are single-word transfer mode, burst mode, and intermittent mode. The transfer mode is selected by setting the DMnTM1 and 0 bits in the DMA control register.

6.4.2.1 Single-word Transfer

One transfer is performed in response to one transfer request. An interrupt request is generated once the number of transfers (1 to 65536) specified in the DMA transfer word count register are completed.
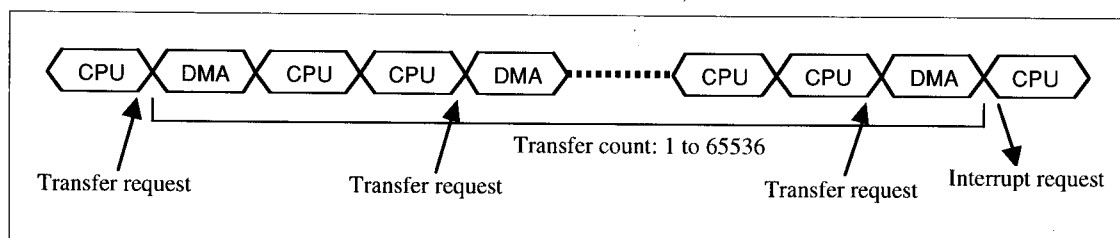
Fig. 6-4-1 Example of Transfer in Single-word Transfer Mode

6.4.2.2 Burst Transfer Mode

The number of transfers specified in the DMA transfer count register are performed in response to one transfer request. The bus is not released until the transfer is completed. Transfers are only interrupted by non-maskable interrupts, external bus requests, and DRAM refresh requests.
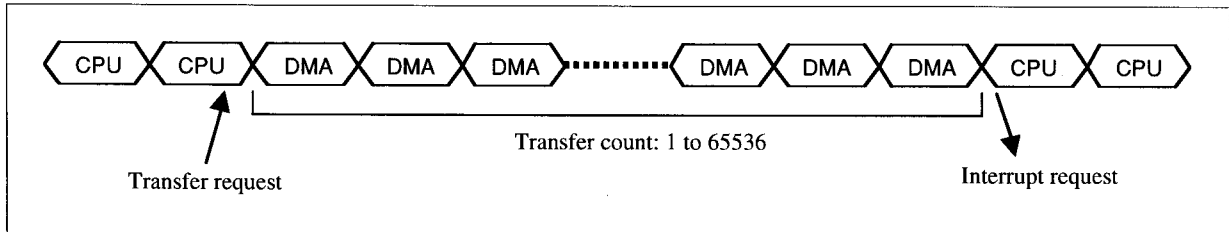
Fig. 6-4-2 Example of Transfer in Burst Transfer Mode

6.4.2.3 Intermittent Transfer Mode

The number of transfers specified in the DMA transfer word count register are performed in response to one transfer request. However, the bus is released once after the number of transfers specified in the DMA transfer word count register are completed. During continuous transfer (during the transfer of the number of intermittent cycles), the bus is not released except because of a non-maskable interrupt, an external bus request, or a DRAM refresh request.
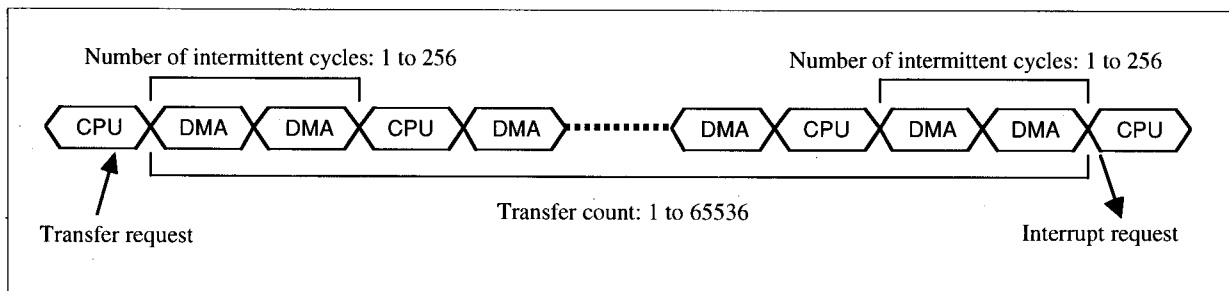
Fig. 6-4-3 Example of Transfer in Intermittent Transfer Mode

6.4.3 Priority Ranking

When there are multiple requests to use the bus, they are processed according to the following rankings:

(Internal bus)   DMA ch0 > DMA ch1 > DMA ch2 > DMA ch3 > CPU

(External bus)   DRAM refresh > External bus request > DMA ch0

> DMA ch1 > DMA ch2 > DMA ch3 > CPU

Processing proceeds as follows when there is a transfer request on a channel with a higher priority while a DMA transfer is in progress.

During single-word transfer: After the single-word transfer is completed, the DMA with the high priority is executed. When the DMA with high priority is executed, the transfer that was interrupted is resumed after the high-priority DMA transfer is completed.

During burst transfer: The high priority DMA is executed after the transfer is completed.

During intermittent transfer: After the specified number of intermittent transfer cycles are completed, the DMA with the high priority is executed. When the DMA with the high priority is executed, the transfer that was interrupted is resumed after the high-priority DMA transfer is completed.

If a DRAM refresh request or an external bus request is generated while a burst transfer or an intermittent transfer is in progress, the transfer is interrupted and the bus is released. The transfer is then resumed after the bus is no longer needed for the DRAM refresh or external device. In addition, if the CPU requests the use of a non-conflicting bus while a burst transfer is in progress (for example, if the CPU attempted to access the external bus while I/O was being accessed during a DMA transfer between I/O and the external bus), then the CPU access cycle may be inserted.

6.4.4 Transfer Start / End / Interrupt

Transfer start

A transfer starts if a startup source set in the DMnBG4 to 0 bits of the control register is generated and the DMnTEN bit is "1".

Transfer end

If a transfer is performed according to the method set in the DMA control register, then the transfer ends when the transfer count set in the DMA transfer word count register reaches "0".

Transfer interrupt

If a transfer is in progress, it is interrupted if a high-priority bus request or DMA transfer request is generated or a non-maskable interrupt is generated.

> **!** For details on priority rankings, refer to section 6.4.3, "Priority Ranking."

6.4.5   Notes on Programming

- Notes on setting DMA

  (1) When enabling DMA, complete all of the DMA-related settings and wait state bus mode settings before writing a "1" to the DMnTEN bit in the DMA control register.

  (2) The DMA settings and bus mode settings cannot be changed while DMA is running. If any of these settings need to be changed, write a "0" to the DMnTEN bit in the DMA control register first.

- Notes about DMA for external memory

  In the case of DMA for external memory, areas that span multiple blocks cannot be specified for the transfer source or the transfer destination.

- Notes about intermittent DMA transfer

  Before operating an intermittent DMA transfer, set the registers, DMA transfer word count register[DMnCNT] and DMA intermittent cycle Register[DMnCYC], to the condition "DMA transfer number = intermittent cycle × N".

  If the numbers of DMA transfer are more than "intermittent transfer cycles × N", then perform the extra transfer by using one of the following methods:

  Method 1:   Generate a DMA transfer end interrupt and have the CPU perform the transfer.

  Method 2:   Set the remaining transfer in burst transfer mode with the same transfer source on a channel with a lower priority than the transfers described above. As a result of doing so, the remaining transfer will be executed as soon as the above transfers are completed.

For details on blocks, refer to section 5.7, "Memory Spaces."

• Notes on the DMA single-word transfer

When the DMA single-word transfer is performed in response to the initiation source of serial I/F, timer or interrupt, sometime the transfer is performed twice for the one request of the transfer .
During the 2nd transfer, the source/destination address will be updated again.
They are calculated with setting in DMAC control register: increment, decrement or specified formula.

The DMA initiation source mentioned above are following:
• serial 0 transmission end source
• serial 0 reception end source
• serial 1 transmission end source
• serial 1 reception end source
• serial 2 transmission end source
• serial 2 reception end source
• timer 2 underflow source
• timer 3 underflow source
• timer 6A compare/capture source
• IRQ0 input source
• IRQ1 input source

When use the above initiation source, set the dummy DMA transmission generated by same request of the original DMA transmission.

The dummy DMA transmission setting is as follows:
• set the dummy DMA before the original DMA setting.
• Use the higher priority channel for the dummy DMA than that of the original DMA.
  For instance, when the original DMA use channel 2, the dummy DMA must use channel 0 or channel 1.
• For the dummy DMA , the transfer addressing modes of source side and destination side (DMnSAM field and DMnDAM field of DMnCTR register) must be fixed.
• The transfer mode (DMnTM field of DMnCTR register) for the dummy DMA must be set as single-word transfer mode.
• For the transfer word register (DMnCNT) of the dummy DMA, give the number, twice of the original transfer word number plus 2.
  (Due to the possibility, another initiate source can be generated during the time difference of the setting the dummy DMA and the original DMA.)
• The source address register (DMnSRC) and the destination address register (DMnDST) for the dummy DMA must be set to the unused address of the internal IO area (for instance 0x38000000).

By setting above mentioned, when there is an initiation source, first,the dummy DMA is performed. After the dummy DMA perform a single-word transmission, the original DMA is activated. The original DMA transfer is correctly performed only once.

- Notes on the execution of a BSET or BCLR instruction.

    Interrupts are prohibited and the bus is locked (occupied by the CPU) when BSET or BCLR is being executed. However, if a BSET or BCLR instruction is executed during program execution in external memory, a bus authority release due to an external bus request or DMA transfer may be interposed between the data read and data write by the BSET or BCLR instruction.

    If the atomic bus cycles of the BSET or BCLR instruction need to be guaranteed in a system that uses multiple processors, either of the following measures should be taken.

    1. A program in which a BSET or BCLR instruction is executed should be placed in instruction cache. Note that an external memory access is occurred, if a cache miss occurs in the instruction cache.

    2. Program so that bus requests cannot be accepted during execution of a BSET or BCLR instruction.

6.4.6 DMA Transfer Examples

6.4.6.1 One Bus Cycle Transfers

One bus cycle transfer (single-word transfer, DMR edge detection, synchronous mode)

Transfer devices: External memory → External device with acknowledge function
Transfer method: One bus cycle, single-word transfer, DMR edge detection
Bus mode: 32-bit bus, synchronous mode, one wait state
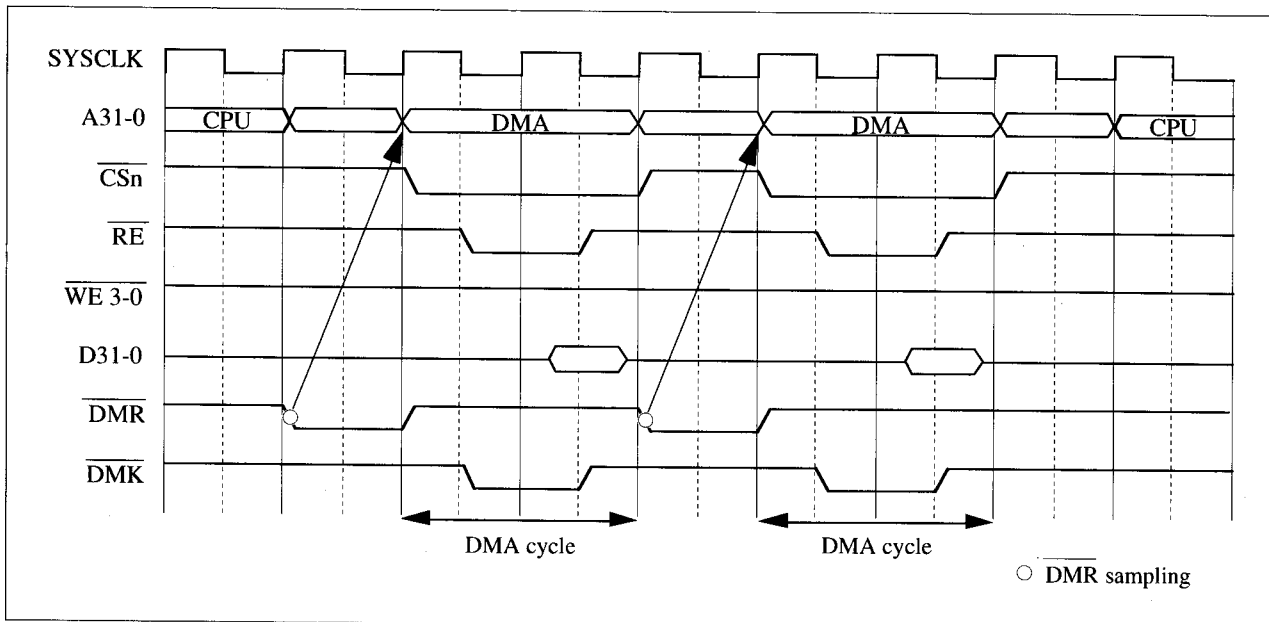
Fig. 6-4-4 One Bus Cycle Transfer (Single-word Transfer, DMR Edge Detection, Synchronous Mode)

Fig. 6-4-4 shows the timing chart for a one bus cycle transfer in synchronous mode from external memory to an external device that supports the acknowledge function, initiated by edge detection of a DMA request signal (DMR) from an external source. With edge detection, the DMR signal is detected at its falling edge. Once a DMA request is detected, detection does not occur until the DMA acknowledge signal (DMK) is asserted.

External memory, the transfer source, is addressed and the data is read at the same time that the external device, the transfer destination, is accessed and written in coordination with the DMK signal. Note that the number of bus cycles and the bus mode for an external memory access are determined by the values set in the memory control register, just as in the case of an external access by the CPU. However, in a one-bus cycle transfer, the bus widths of external memory and the external device must be the same.

One bus cycle transfer (burst transfer, DMR edge detection, synchronous mode)

Transfer devices:	External device with acknowledge function → External memory
Transfer method:	One bus cycle, burst transfer, DMR edge detection
Bus mode:	32-bit bus, synchronous mode, one wait state
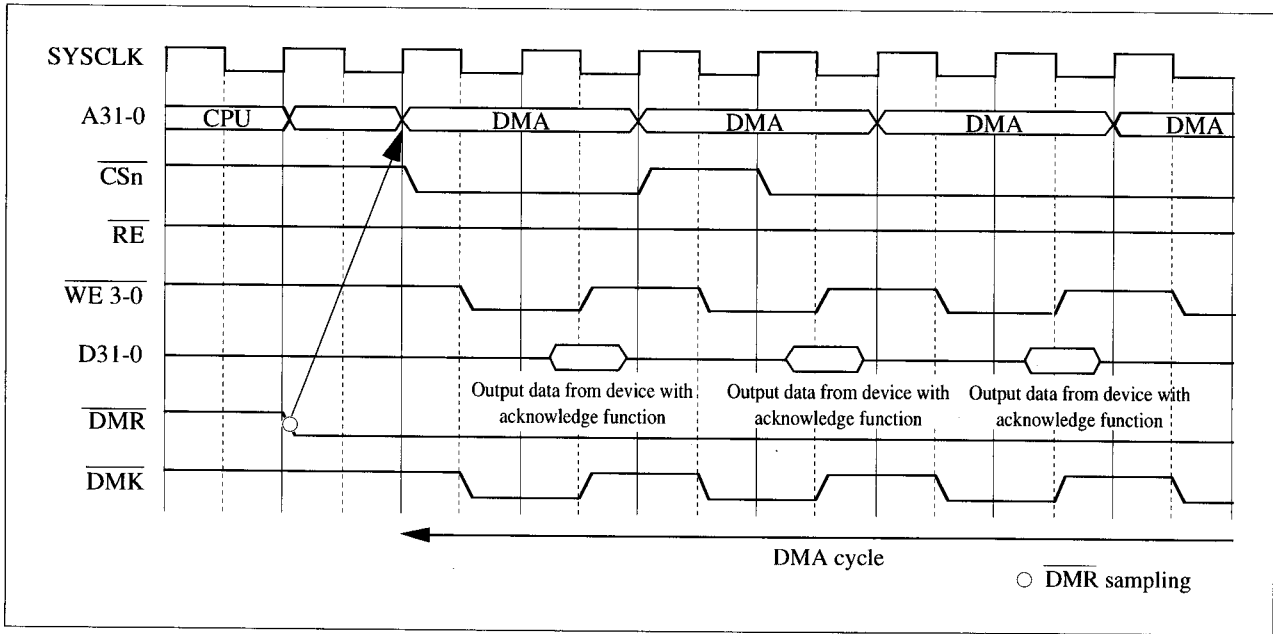
Fig. 6-4-5 One Bus Cycle Transfer (Burst Transfer, DMR Edge Detection, Synchronous Mode)

Fig. 6-4-5 shows the timing chart for a burst transfer in synchronous mode from an external device that supports the acknowledge function to external memory, initiated by edge detection of a DMA request signal (DMR) from an external source. Once the edge is detected, detection does not occur again until the burst transfer is completed.

One bus cycle transfer (single-word transfer, DMR edge detection, asynchronous mode)

Transfer devices: External memory → External device with acknowledge function

Transfer method: One bus cycle, single-word transfer, DMR edge detection

Bus mode: 32-bit bus, asynchronous mode, one wait state
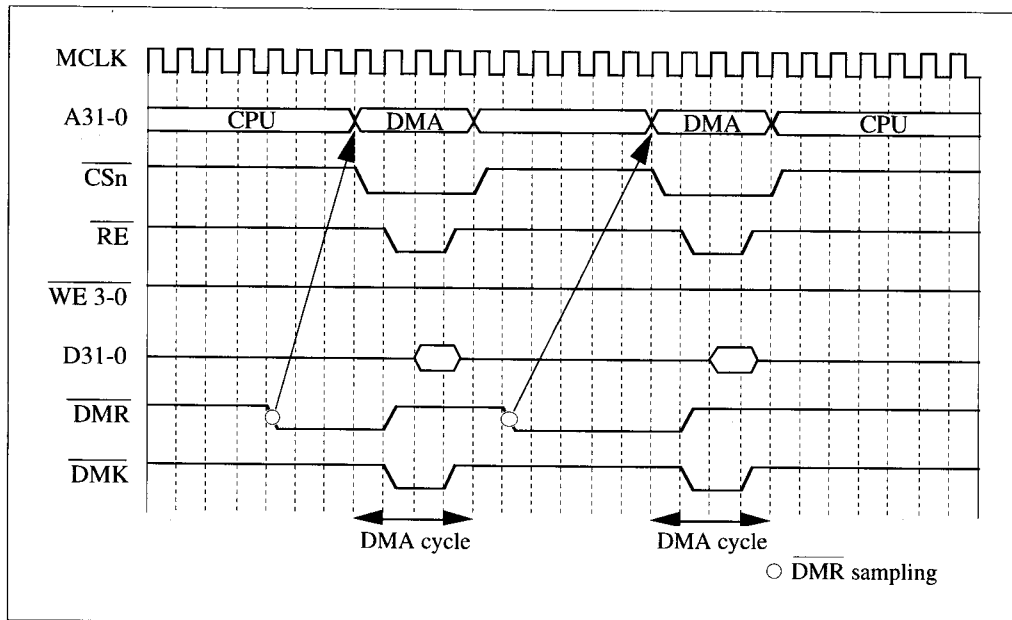
Fig. 6-4-6 One Bus Cycle Transfer (Single-word Transfer, DMR Edge Detection, Asynchronous Mode)
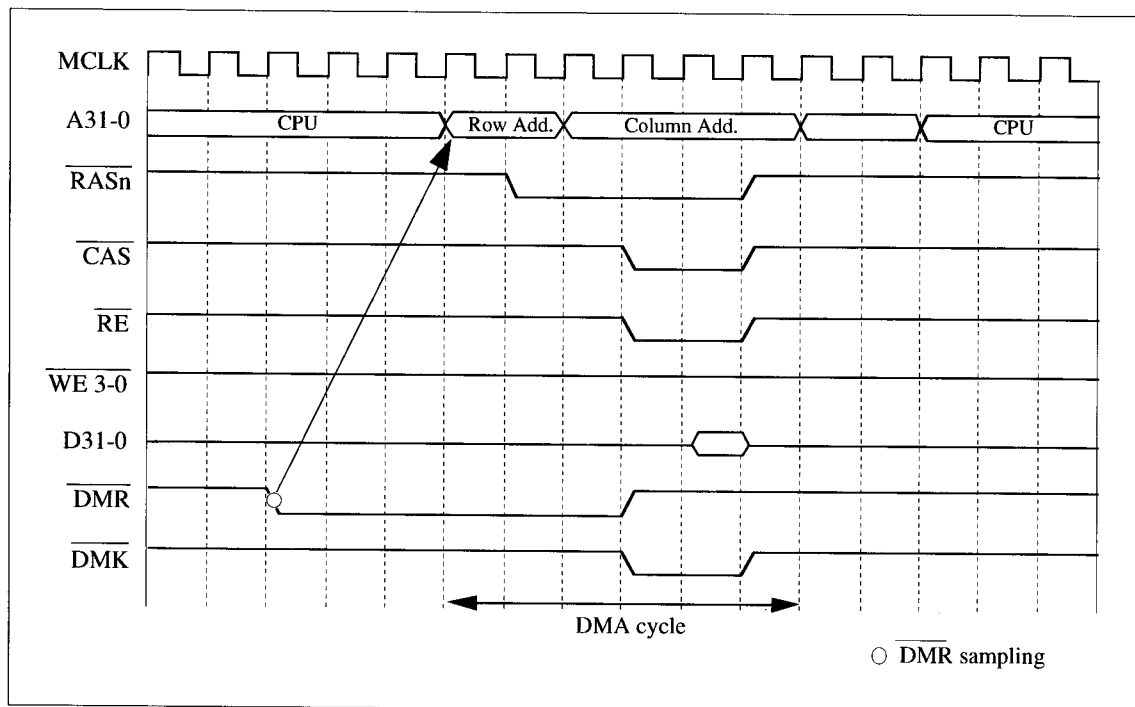
Fig. 6-4-7 One Bus Cycle Transfer (Single-word Transfer, DMR Edge Detection, Asynchronous Mode)

Figs. 6-4-6 and 6-4-7 show the timing charts for a one bus cycle transfer in asynchronous mode from external memory (a non-DRAM space in Fig. 6-4-6; a DRAM space in Fig. 6-4-7) to an external device that supports the acknowledge function, initiated by edge detection of a DMA request signal (DMR) from an external source. With edge detection, the DMR signal is detected at its falling edge, but once a DMA request is detected, detection does not occur again until the DMA acknowledge signal (DMK) is asserted.

6.4.6.2 Two Bus Cycle Transfers

Two bus cycle transfers (burst transfer, DMR edge detection, synchronous mode)

Transfer devices: External memory → External memory

Transfer method: Two bus cycles, burst transfer, DMR edge detection

Bus mode: Transfer source: 32-bit bus, synchronous mode, 1 wait state

Transfer destination: 32-bit bus, synchronous mode, 1 wait state
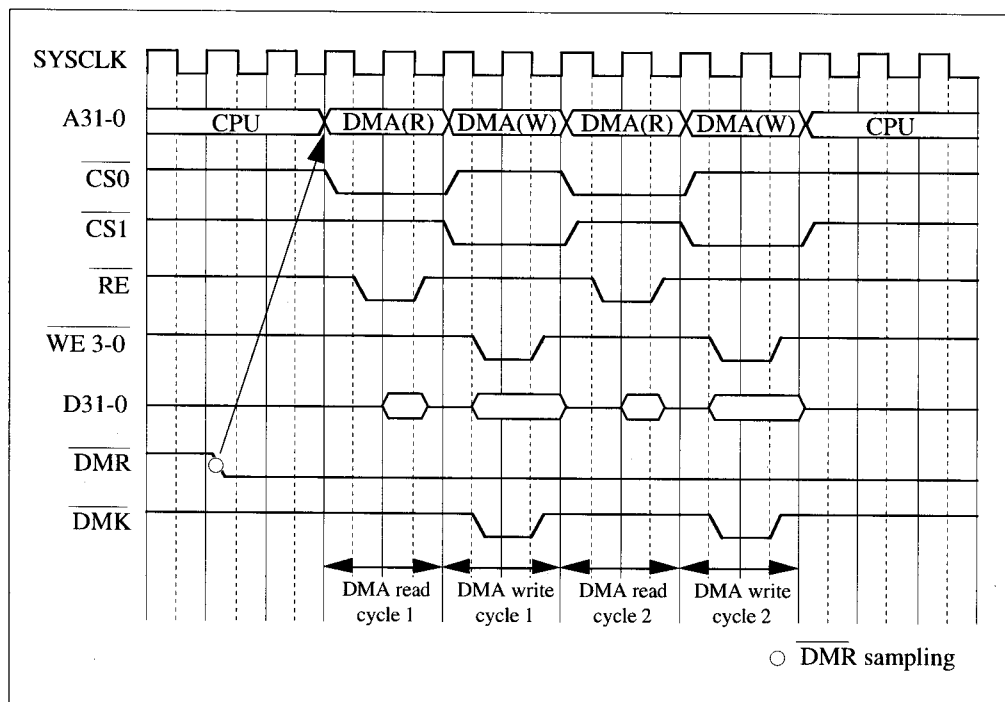
Fig. 6-4-8 Two Bus Cycle Transfer (Burst Transfer, DMR Edge Detection, Synchronous Mode)

Fig. 6-4-8 shows the timing chart for a two bus cycle burst transfer in synchronous mode from external memory to external memory, initiated by edge detection of a DMA request signal (DMR) from an external source. In a two bus cycle transfer initiated by the DMR signal, the acknowledge signal (DMK) is asserted during the write transfer cycle of the DMA transfer.

Note that the number of bus cycles and the bus mode for external memory access are determined by the values set in the memory control register, just as in the case of external accesses performed by the CPU. Transfer between devices with different bus widths is possible.

Two bus cycle transfer (single-word transfer, serial reception end interrupt)

Transfer devices: Serial reception buffer → External memory

Transfer method: Two bus cycles, single-word transfer, serial reception end interrupt

Bus mode: Transfer source: 32-bit bus, synchronous mode, 0 wait states

Transfer destination: 32-bit bus, asynchronous mode, 3 wait states
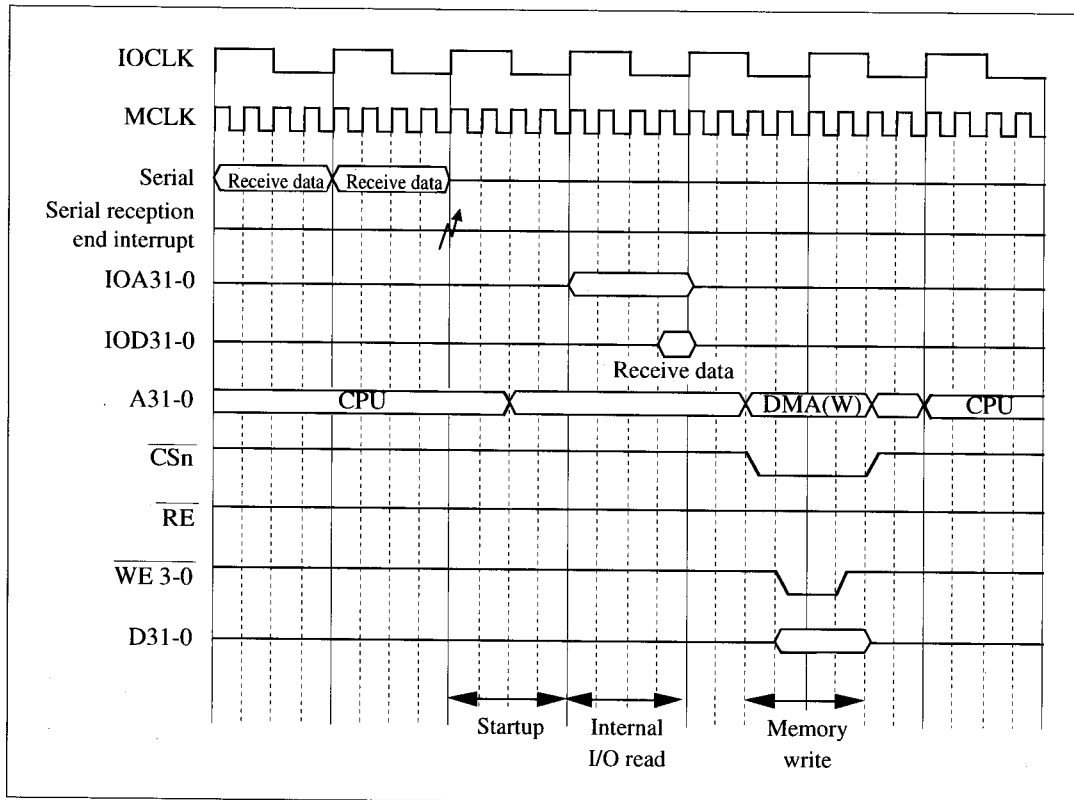
Fig. 6-4-9 Two Bus Cycle Transfer (Single-word Transfer, Serial Reception End Interrupt)

Fig. 6-4-9 shows the timing chart for a two bus cycle transfer from the serial reception buffer to external memory, initiated by a serial reception end interrupt. The DMA transfer begins within a minimum of four CPU cycles after the serial reception end interrupt is input to the DMA. After the contents of the serial reception data register are read into the DMA data buffer via I/O bus access, then the external bus cycle starts within a minimum of two CPU cycles and the data is written to external memory. Note that the number of bus cycles and the bus mode for external memory access are determined by the values set in the memory control register, just as in the case of external accesses performed by the CPU.

Chapter 7. Interrupt Controller

7

7.1 Overview

The interrupt controller processes non-maskable interrupts and level interrupts (internal interrupts and external interrupts).

The microcontroller has eight external interrupt pins and one non-maskable interrupt pin. External level interrupts are processed when an external pin interrupt signal is maintained for two or more I/O clock cycles.

7.2 Features

- Up to four interrupt requests can be accepted by each group.
- Interrupt mask level
  Can be set for each interrupt group.
- External pin interrupt conditions
  Positive edge, negative edge, high level, low level
  Recovery from STOP, HALT, or SLEEP mode is possible by means of an external pin interrupt

7.3 Configuration

7.3.1 System Diagram
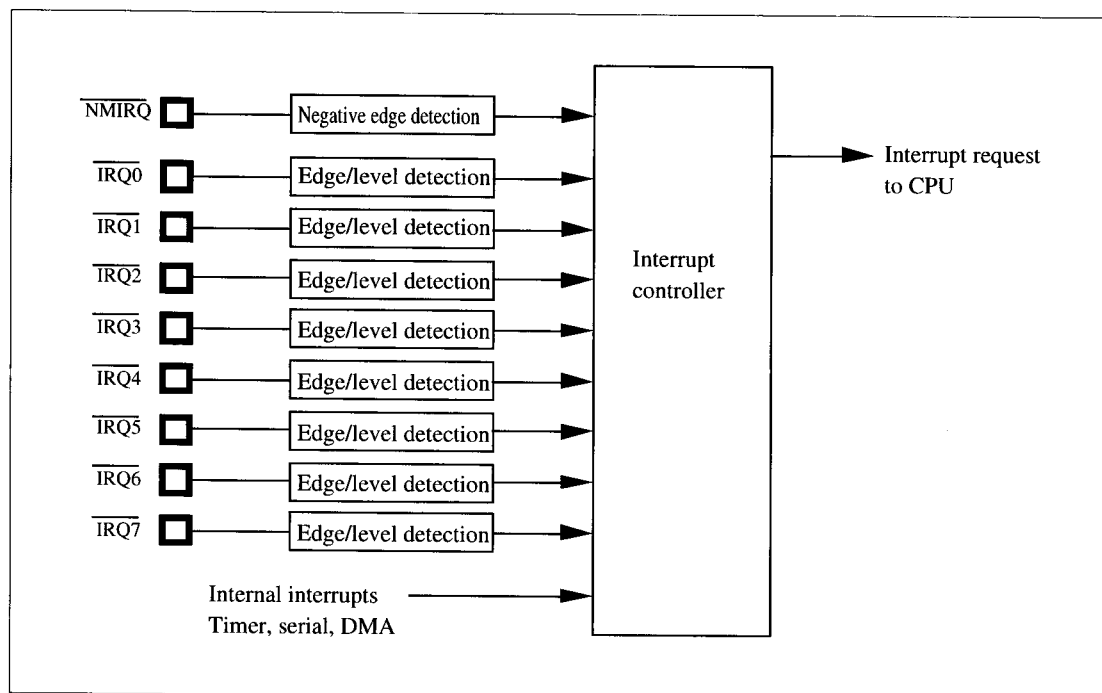
Fig. 7-3-1 System Diagram

7.3.2　Interrupt Source Assignments

The interrupt level can be set independently for each group. (However, GROUP0 and GROUP1 are non-maskable.)
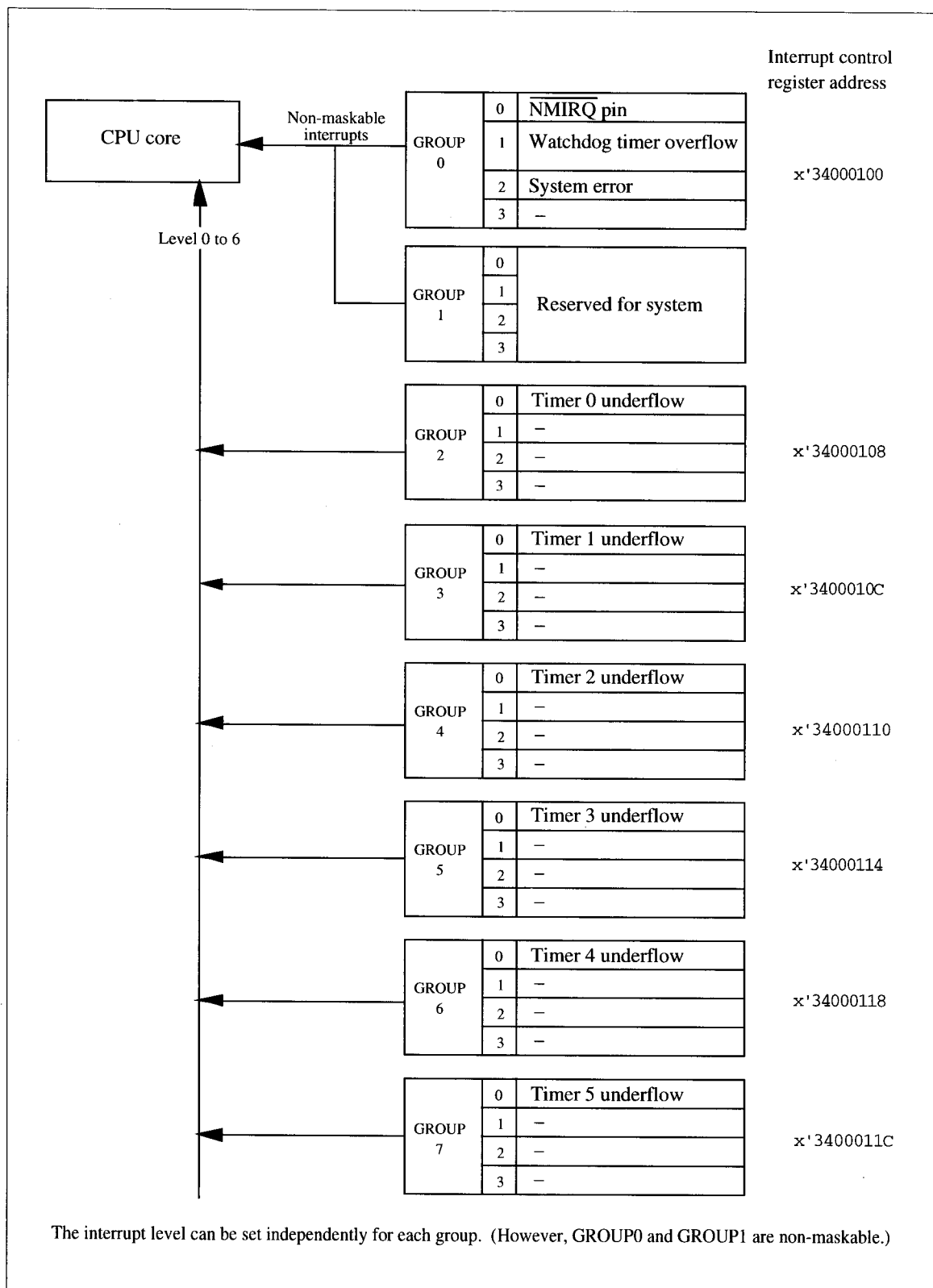
Fig. 7-3-2　Block Diagram 1
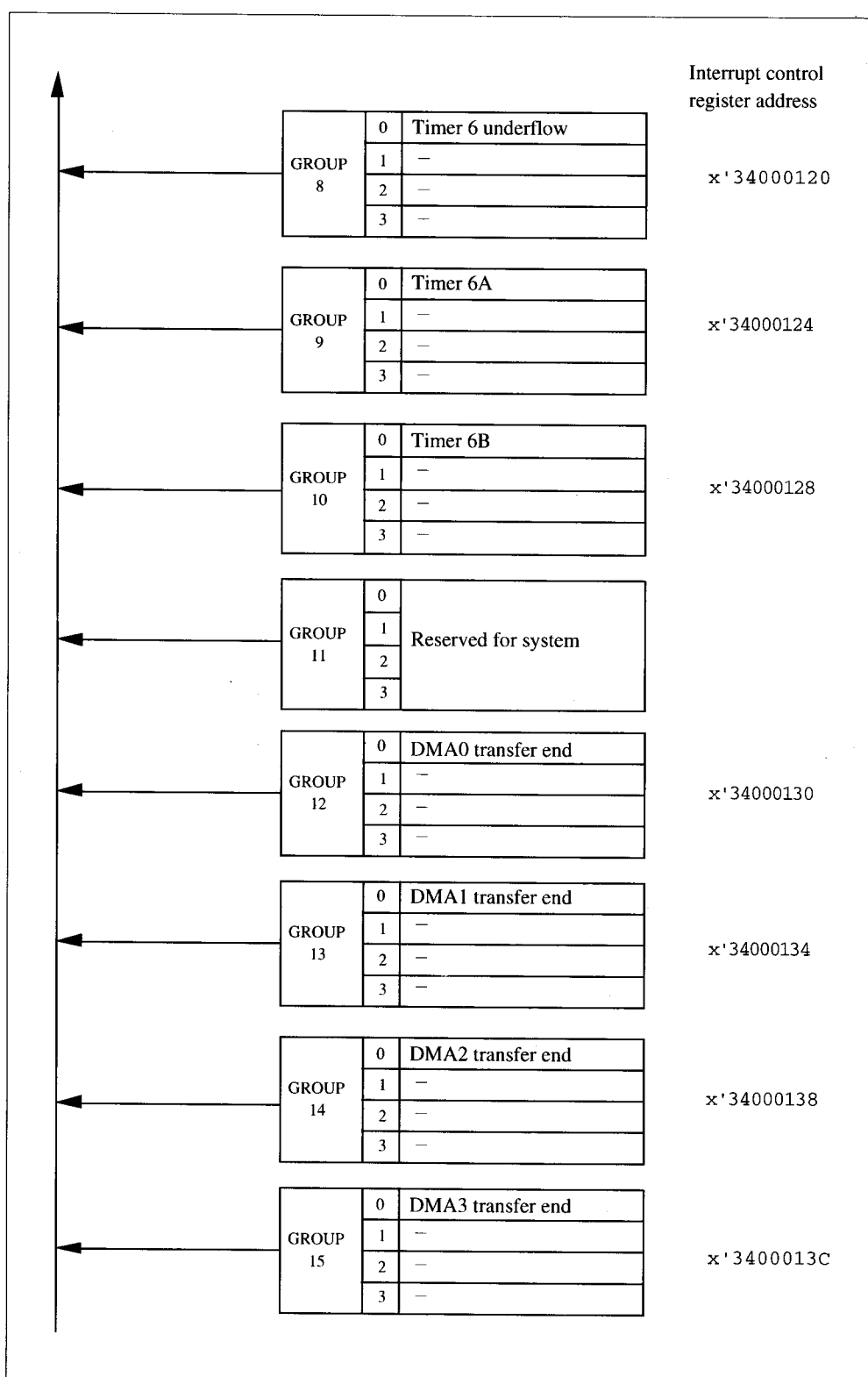
Fig. 7-3-3 Block Diagram 2
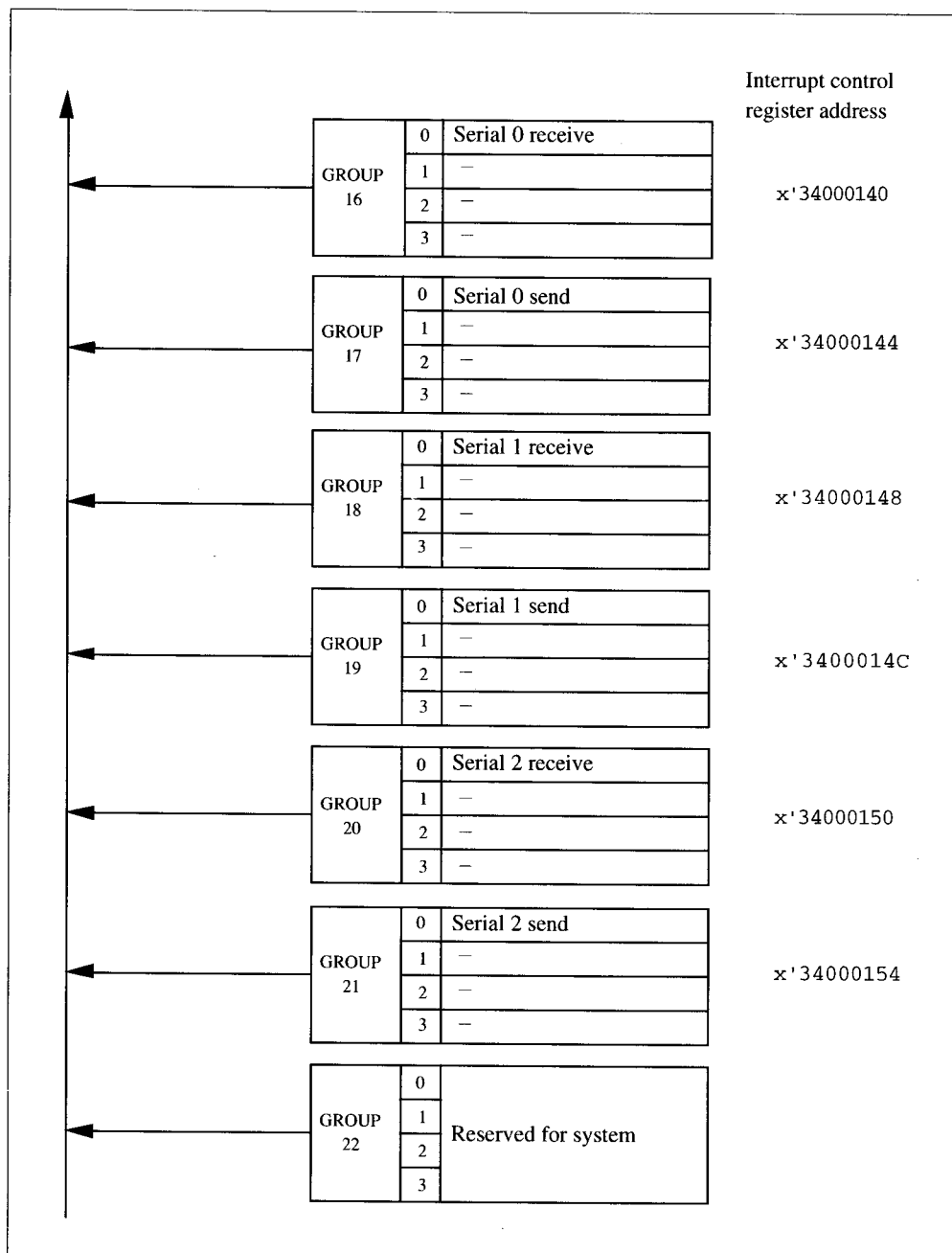
Fig. 7-3-4 Block Diagram 3
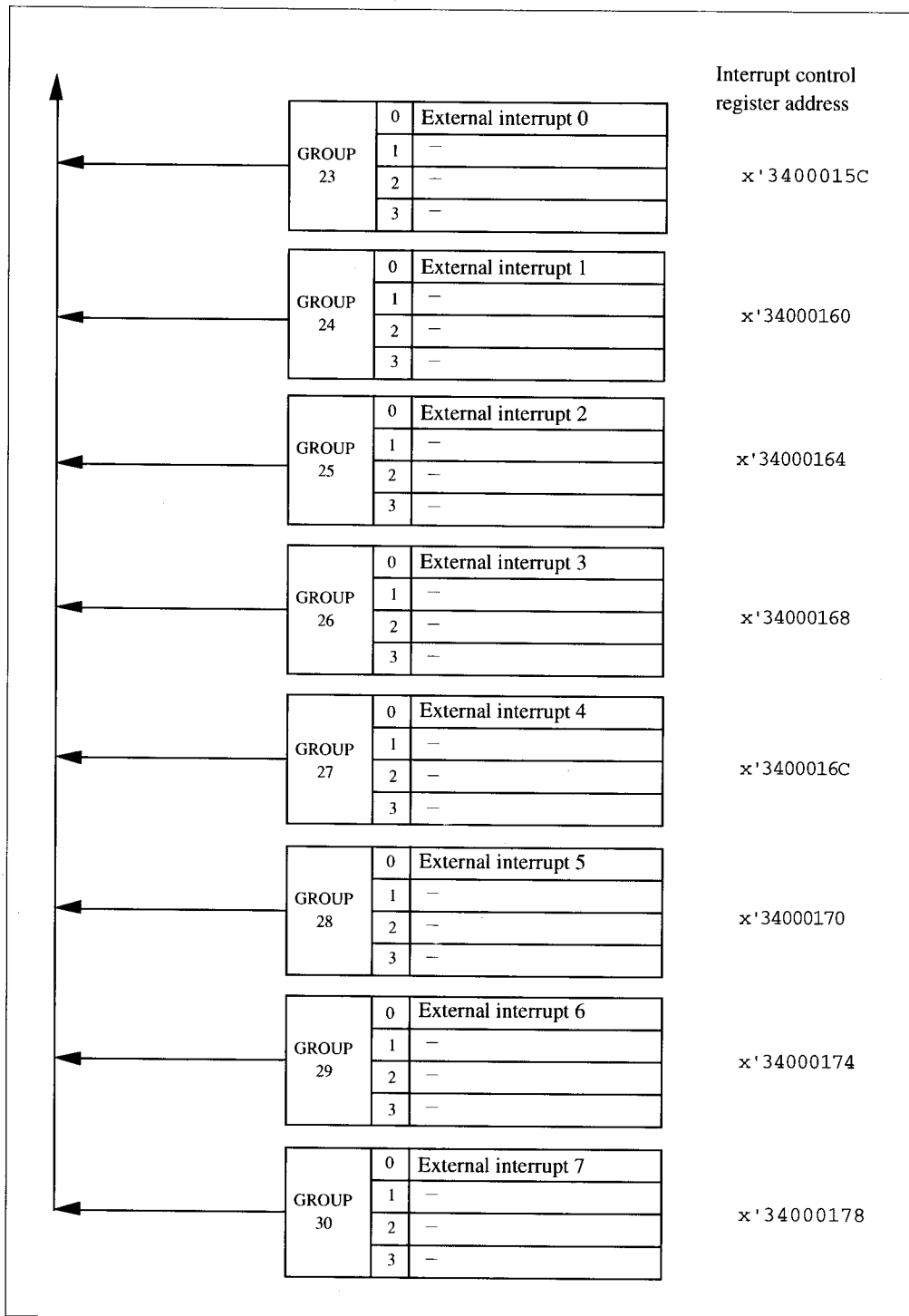
Fig. 7-3-5 Block Diagram 4

7.4   Description of Registers

This interrupt controller includes interrupt control registers, an interrupt acceptance group register, and an external interrupt condition specification register.

Register list   The registers are described on the pages that follow.

Table 7-4-1 Register List

Address	Register name	Symbol	Number of bits	Initial value	Access size
x'34000100	Non-maskable interrupt control register	G0ICR(NMICR)	16	x'0000	8,16
x'34000108	Group 2 interrupt control register	G2ICR	16	x'0000	8,16
x'3400010C	Group 3 interrupt control register	G3ICR	16	x'0000	8,16
x'34000110	Group 4 interrupt control register	G4ICR	16	x'0000	8,16
x'34000114	Group 5 interrupt control register	G5ICR	16	x'0000	8,16
x'34000118	Group 6 interrupt control register	G6ICR	16	x'0000	8,16
x'3400011C	Group 7 interrupt control register	G7ICR	16	x'0000	8,16
x'34000120	Group 8 interrupt control register	G8ICR	16	x'0000	8,16
x'34000124	Group 9 interrupt control register	G9ICR	16	x'0000	8,16
x'34000128	Group 10 interrupt control register	G10ICR	16	x'0000	8,16
x'34000130	Group 12 interrupt control register	G12ICR	16	x'0000	8,16
x'34000134	Group 13 interrupt control register	G13ICR	16	x'0000	8,16
x'34000138	Group 14 interrupt control register	G14ICR	16	x'0000	8,16
x'3400013C	Group 15 interrupt control register	G15ICR	16	x'0000	8,16
x'34000140	Group 16 interrupt control register	G16ICR	16	x'0000	8,16
x'34000144	Group 17 interrupt control register	G17ICR	16	x'0000	8,16
x'34000148	Group 18 interrupt control register	G18ICR	16	x'0000	8,16
x'3400014C	Group 19 interrupt control register	G19ICR	16	x'0000	8,16
x'34000150	Group 20 interrupt control register	G20ICR	16	x'0000	8,16
x'34000154	Group 21 interrupt control register	G21ICR	16	x'0000	8,16
x'3400015C	Group 23 interrupt control register	G23ICR	16	x'0000	8,16
x'34000160	Group 24 interrupt control register	G24ICR	16	x'0000	8,16
x'34000164	Group 25 interrupt control register	G25ICR	16	x'0000	8,16
x'34000168	Group 26 interrupt control register	G26ICR	16	x'0000	8,16
x'3400016C	Group 27 interrupt control register	G27ICR	16	x'0000	8,16
x'34000170	Group 28 interrupt control register	G28ICR	16	x'0000	8,16
x'34000174	Group 29 interrupt control register	G29ICR	16	x'0000	8,16
x'34000178	Group 30 interrupt control register	G30ICR	16	x'0000	8,16
x'34000200	Interrupt acceptance group register	IAGR	16	x'0000	8,16
x'34000280	External interrupt condition specification register	EXTMD	16	x'0000	8,16

7.4.1 Non-maskable Interrupt Control Register

Register symbol: G0ICR (NMICR)

Address: x' 34000100

Purpose: Governs the generation of a non-maskable interrupt.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	-	-	-	-	SYS EF	WDIF	NMIF
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	NMIF	External non-maskable interrupt request flag
		0: Off 1: On
1	WDIF	Watchdog timer overflow interrupt request flag
		0: Off 1: On
2	SYSEF	System error interrupt request flag
		0: Off 1: On

The method of clearing flag differs according to the interrupt request flags.

1. External non-maskable interrupt request flag (NMIF) and Watchdog timer overflow interrupt request flag (WDIF)

    After a non-maskable interrupt is accepted, these flags can be cleared by writing to the non-maskable interrupt control register (NMICR).

    When a flag is set to "1", write a "1" to the flag to clear it.

    The relationship between the flag status, the data written to the flag, and the new flag status after the data is written is shown in the table below.

Flag status	Write data	Flag status after write	Change in flag
0	0	0	No change
0	1	0	No change
1	0	1	No change
1	1	0	Flag is cleared

2. System error interrupt request flag (SYSEF)

    This flag cannot be cleared by writing to the non-maskable interrupt control register (NMICR).

    This flag can be cleared by generating a reset interrupt by setting the $\overline{\text{RST}}$ pin to "L" level or by the self-reset, which is generated by writing to the reset control register (RSTCTR) of the watchdog timer.

Non-maskable interrupts cannot be generated by software.

7.4.2 Group n Interrupt Control Registers GnICR (n = 2 to 30)

These registers are used to control level interrupts in each group from 2 to 30. This register confirms the group interrupt level as well as the enabling, request, and detection of interrupts within the group.

This page describes the common elements of G2ICR through G30ICR.

The level interrupt control registers for group 2 to group 30 are described individually on the pages that follow.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	LV2	LV1	LV0	IE3	IE2	IE1	IE0	IR3	IR2	IR1	IR0	ID3	ID2	ID1	ID0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0 to 3	ID0 to ID3	Group n interrupt detection register

- This register stores the logical product of the IE and IR bits.
- If an interrupt that is enabled by the IE bits is generated, the ID bit corresponding to the interrupt is set to "1".

4 to 7	IR0 to IR3	Group n interrupt request register

- This register stores interrupt requests. Each bit corresponds to an interrupt.
- After receiving the interrupt, these bits are cleared by software in the interrupt processing program.
- When clearing one of these bits, write a "0" to the bit to be cleared and write a "1" to the corresponding bit from ID0 to 3.

Write data		Result of write	
I R	I D	I R	I D
0	0	No change	No change
1	0	No change	No change
0	1	0	0
1	1	1	Value of IE

8 to 11	IE0 to IE3	Group n interrupt enable register

- This register is used to specify whether an interrupt is enabled or not.
- When an IE bit is set to "1", the corresponding interrupt is enabled.
- Setting an IE bit while the corresponding IR bit is set generates an interrupt.

<Continued>

<Continued>

Bit No.	Bit name	Description
12 to 14	LV0 to LV2	Group n interrupt priority level register

* This register sets the interrupt priority levels.
* If the interrupt level set in the LV (2:0) bits is smaller than the IM (2:0) bits in the PSW, interrupts of the corresponding interrupt group are permitted. Interrupts in the same interrupt group are all of the level specified by the LV (2:0) bits.
* When simultaneous interrupt requests are generated from more than one interrupt group, the interrupt with the highest interrupt priority level is accepted. In addition, if multiple interrupt priority levels are set in the same level, the interrupt from the group with the smallest group number is accepted.

For details on the interrupt sources assigned to each group, refer section 7.3.2, "Interrupt Source Assignments."

Changing values in LV2 to LV0, interrut priorty level register of GnICR, or in IE3 to IE0, interrupt enable register of GnICR, must be operated under the condition, inhibiting any interruption.

and	0xf7ff, psw	; clears IE of PSW
nop		; makes it sure that IE to be cleared
nop		; in pipe line process
mov	d0, (GnICR)	; sets LV2 to LV0, and/or IE3 to IE0
mov	(GnICR), psw	; synchronizes with the store buffer
or	0x0800, psw	; sets IE of PSW

While the interrupt handler is being operated, IE equals to 0 unless IE has been set. So it's not necessary to clear IE.

Nop instruction, shown in above example, can be replaced with any instruction unless it causes the changing the values in LV2 to LV0, and/or in IE3 to IE0.

As shown in above example, nop is inserted twice. This is just to make it sure to have minimum cycles need to change the value in IE of PSW. I.e. any instruction set, require more cycle than nop instruction twice, can be used.

7.4.2.1 Group 2 Interrupt Control Register

Register symbol: G2ICR

Address: x' 34000108

Purpose: This register is used to enable group 2 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G2 LV2	G2 LV1	G2 LV0	-	-	-	TM0 IE	-	-	-	TM0 IR	-	-	-	TM0 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM0ID	Timer 0 underflow interrupt detection flag
		0: No underflow interrupt detected
		1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM0IR	Timer 0 underflow interrupt request flag
		0: No underflow interrupt requested
		1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM0IE	Timer 0 underflow interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G2LV0	Group 2 interrupt priority level register (LSB)
13	G2LV1	Group 2 interrupt priority level register
14	G2LV2	Group 2 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.2 Group 3 Interrupt Control Register

Register symbol: G3ICR

Address: x' 3400010C

Purpose: This register is used to enable group 3 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G3 LV2	G3 LV1	G3 LV0	-	-	-	TM1 IE	-	-	-	TM1 IR	-	-	-	TM1 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM1ID	Timer 1 underflow interrupt detection flag 0: No underflow interrupt detected 1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM1IR	Timer 1 underflow interrupt request flag 0: No underflow interrupt requested 1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM1IE	Timer 1 underflow interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G3LV0	Group 3 interrupt priority level register (LSB)
13	G3LV1	Group 3 interrupt priority level register
14	G3LV2	Group 3 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.3   Group 4 Interrupt Control Register

Register symbol: G4ICR

Address: x' 34000110

Purpose: This register is used to enable group 4 interrupts, and to confirm interrupt requests
                      and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G4 LV2	G4 LV1	G4 LV0	-	-	-	TM2 IE	-	-	-	TM2 IR	-	-	-	TM2 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM2ID	Timer 2 underflow interrupt detection flag 0: No underflow interrupt detected 1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM2IR	Timer 2 underflow interrupt request flag 0: No underflow interrupt requested 1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM2IE	Timer 2 underflow interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G4LV0	Group 4 interrupt priority level register (LSB)
13	G4LV1	Group 4 interrupt priority level register
14	G4LV2	Group 4 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.4 Group 5 Interrupt Control Register

Register symbol: G5ICR

Address: x' 34000114

Purpose: This register is used to enable group 5 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G5 LV2	G5 LV1	G5 LV0	-	-	-	TM3 IE	-	-	-	TM3 IR	-	-	-	TM3 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM3ID	Timer 3 underflow interrupt detection flag 0: No underflow interrupt detected 1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM3IR	Timer 3 underflow interrupt request flag 0: No underflow interrupt requested 1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM3IE	Timer 3 underflow interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G5LV0	Group 5 interrupt priority level register (LSB)
13	G5LV1	Group 5 interrupt priority level register
14	G5LV2	Group 5 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.5   Group 6 Interrupt Control Register

Register symbol: G6ICR

Address: x' 34000118

Purpose: This register is used to enable group 6 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G6 LV2	G6 LV1	G6 LV0	-	-	-	TM4 IE	-	-	-	TM4 IR	-	-	-	TM4 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM4ID	Timer 4 underflow interrupt detection flag
		0: No underflow interrupt detected
		1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM4IR	Timer 4 underflow interrupt request flag
		0: No underflow interrupt requested
		1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM4IE	Timer 4 underflow interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G6LV0	Group 6 interrupt priority level register (LSB)
13	G6LV1	Group 6 interrupt priority level register
14	G6LV2	Group 6 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.6 Group 7 Interrupt Control Register

Register symbol: G7ICR

Address: x' 3400011C

Purpose: This register is used to enable group 7 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G7 LV2	G7 LV1	G7 LV0	-	-	-	TM5 IE	-	-	-	TM5 IR	-	-	-	TM5 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM5ID	Timer 5 underflow interrupt detection flag 0: No underflow interrupt detected 1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM5IR	Timer 5 underflow interrupt request flag 0: No underflow interrupt requested 1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM5IE	Timer 5 underflow interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G7LV0	Group 7 interrupt priority level register (LSB)
13	G7LV1	Group 7 interrupt priority level register
14	G7LV2	Group 7 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.7 Group 8 Interrupt Control Register

Register symbol: G8ICR

Address: x' 34000120

Purpose: This register is used to enable group 8 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G8 LV2	G8 LV1	G8 LV0	-	-	-	TM6 IE	-	-	-	TM6 IR	-	-	-	TM6 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	TM6ID	Timer 6 underflow interrupt detection flag 0: No underflow interrupt detected 1: Underflow interrupt detected
1 to 3	—	Always returns "0".
4	TM6IR	Timer 6 underflow interrupt request flag 0: No underflow interrupt requested 1: Underflow interrupt requested
5 to 7	—	Always returns "0".
8	TM6IE	Timer 6 underflow interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G8LV0	Group 8 interrupt priority level register (LSB)
13	G8LV1	Group 8 interrupt priority level register
14	G8LV2	Group 8 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.8   Group 9 Interrupt Control Register

Register symbol: G9ICR

Address: x' 34000124

Purpose: This register is used to enable group 9 interrupts, and to confirm interrupt requests
                      and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G9 LV2	G9 LV1	G9 LV0	-	-	-	T6A IE	-	-	-	T6A IR	-	-	-	T6A ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	T6AID	Timer 6A interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	T6AIR	Timer 6A interrupt request flag
		0: No interrupt requested
		1: Interrupt requested ·
5 to 7	—	Always returns "0".
8	T6AIE	Timer 6A interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G9LV0	Group 9 interrupt priority level register (LSB)
13	G9LV1	Group 9 interrupt priority level register
14	G9LV2	Group 9 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.9   Group 10 Interrupt Control Register

Register symbol: G10ICR

Address: x' 34000128

Purpose: This register is used to enable group 10 interrupts, and to confirm interrupt re quests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G10 LV2	G10 LV1	G10 LV0	-	-	-	T6B IE	-	-	-	T6B IR	-	-	-	T6B ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	T6BID	Timer 6B interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	T6BIR	Timer 6B interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	T6BIE	Timer 6B interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G10LV0	Group 10 interrupt priority level register (LSB)
13	G10LV1	Group 10 interrupt priority level register
14	G10LV2	Group 10 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.10 Group 12 Interrupt Control Register

Register symbol: G12ICR

Address: x' 34000130

Purpose: This register is used to enable group 12 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G12 LV2	G12 LV1	G12 LV0	-	-	-	DM0 IE	-	-	-	DM0 IR	-	-	-	DM0 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	DM0ID	DMA0 transfer end interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	DM0IR	DMA0 transfer end interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	DM0IE	DMA0 transfer end interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G12LV0	Group 12 interrupt priority level register (LSB)
13	G12LV1	Group 12 interrupt priority level register
14	G12LV2	Group 12 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.11 Group 13 Interrupt Control Register

Register symbol: G13ICR

Address: x' 34000134

Purpose: This register is used to enable group 13 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G13 LV2	G13 LV1	G13 LV0	-	-	-	DM1 IE	-	-	-	DM1 IR	-	-	-	DM1 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	DM1ID	DMA1 transfer end interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	DM1IR	DMA1 transfer end interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	DM1IE	DMA1 transfer end interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G13LV0	Group 13 interrupt priority level register (LSB)
13	G13LV1	Group 13 interrupt priority level register
14	G13LV2	Group 13 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.12 Group 14 Interrupt Control Register

Register symbol: G14ICR

Address: x' 34000138

Purpose: This register is used to enable group 14 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G14 LV2	G14 LV1	G14 LV0	-	-	-	DM2 IE	-	-	-	DM2 IR	-	-	-	DM2 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	DM2ID	DMA2 transfer end interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	DM2IR	DMA2 transfer end interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	DM2IE	DMA2 transfer end interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G14LV0	Group 14 interrupt priority level register (LSB)
13	G14LV1	Group 14 interrupt priority level register
14	G14LV2	Group 14 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.13 Group 15 Interrupt Control Register

Register symbol: G15ICR

Address: x' 3400013C

Purpose: This register is used to enable group 15 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G15 LV2	G15 LV1	G15 LV0	-	-	-	DM3 IE	-	-	-	DM3 IR	-	-	-	DM3 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	DM3ID	DMA3 transfer end interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	DM3IR	DMA3 transfer end interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	DM3IE	DMA3 transfer end interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G15LV0	Group 15 interrupt priority level register (LSB)
13	G15LV1	Group 15 interrupt priority level register
14	G15LV2	Group 15 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.14 Group 16 Interrupt Control Register

Register symbol: G16ICR

Address: x' 34000140

Purpose: This register is used to enable group 16 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G16 LV2	G16 LV1	G16 LV0	-	-	-	SC0R IE	-	-	-	SC0R IR	-	-	-	SC0R ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC0RID	Serial 0 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC0RIR	Serial 0 reception interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC0RIE	Serial 0 reception interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G16LV0	Group 16 interrupt priority level register (LSB)
13	G16LV1	Group 16 interrupt priority level register
14	G16LV2	Group 16 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.15 Group 17 Interrupt Control Register

Register symbol: G17ICR

Address: x' 34000144

Purpose: This register is used to enable group 17 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G17 LV2	G17 LV1	G17 LV0	-	-	-	SC0T IE	-	-	-	SC0T IR	-	-	-	SC0T ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC0TID	Serial 0 transmission interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC0TIR	Serial 0 transmission interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC0TIE	Serial 0 transmission interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G17LV0	Group 17 interrupt priority level register (LSB)
13	G17LV1	Group 17 interrupt priority level register
14	G17LV2	Group 17 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.16 Group 18 Interrupt Control Register

Register symbol: G18ICR

Address: x' 34000148

Purpose: This register is used to enable group 18 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G18 LV2	G18 LV1	G18 LV0	-	-	-	SC1R IE	-	-	-	SC1R IR	-	-	-	SC1R ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC1RID	Serial 1 reception interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC1RIR	Serial 1 reception interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC1RIE	Serial 0 reception interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G18LV0	Group 18 interrupt priority level register (LSB)
13	G18LV1	Group 18 interrupt priority level register
14	G18LV2	Group 18 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.17 Group 19 Interrupt Control Register

Register symbol: G19ICR

Address: x' 3400014C

Purpose: This register is used to enable group 19 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G19 LV2	G19 LV1	G19 LV0	-	-	-	SC1T IE	-	-	-	SC1T IR	-	-	-	SC1T ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC1TID	Serial 1 transmission interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC1TIR	Serial 1 transmission interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC1TIE	Serial 1 transmission interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G19LV0	Group 19 interrupt priority level register (LSB)
13	G19LV1	Group 19 interrupt priority level register
14	G19LV2	Group 19 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.18 Group 20 Interrupt Control Register

Register symbol: G20ICR

Address: x' 34000150

Purpose: This register is used to enable group 20 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G20 LV2	G20 LV1	G20 LV0	-	-	-	SC2R IE	-	-	-	SC2R IR	-	-	-	SC2R ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC2RID	Serial 2 reception interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC2RIR	Serial 2 reception interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC2RIE	Serial 2 reception interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G20LV0	Group 20 interrupt priority level register (LSB)
13	G20LV1	Group 20 interrupt priority level register
14	G20LV2	Group 20 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.19 Group 21 Interrupt Control Register

Register symbol: G21ICR

Address: x' 34000154

Purpose: This register is used to enable group 21 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G21 LV2	G21 LV1	G21 LV0	-	-	-	SC2T IE	-	-	-	SC2T IR	-	-	-	SC2T ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	SC2TID	Serial 2 transmission interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	SC2TIR	Serial 2 transmission interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	SC2TIE	Serial 2 transmission interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G21LV0	Group 21 interrupt priority level register (LSB)
13	G21LV1	Group 21 interrupt priority level register
14	G21LV2	Group 21 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.20 Group 23 Interrupt Control Register

Register symbol: G23ICR

Address: x' 3400015C

Purpose: This register is used to enable group 23 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G23 LV2	G23 LV1	G23 LV0	-	-	-	IQ0 IE	-	-	-	IQ0 IR	-	-	-	IQ0 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ0ID	External interrupt 0 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ0IR	External interrupt 0 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ0IE	External interrupt 0 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G23LV0	Group 23 interrupt priority level register (LSB)
13	G23LV1	Group 23 interrupt priority level register
14	G23LV2	Group 23 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.21 Group 24 Interrupt Control Register

Register symbol: G24ICR

Address: x' 34000160

Purpose: This register is used to enable group 24 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G24 LV2	G24 LV1	G24 LV0	-	-	-	IQ1 IE	-	-	-	IQ1 IR	-	-	-	IQ1 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ1ID	External interrupt 1 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ1IR	External interrupt 1 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ1IE	External interrupt 1 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G24LV0	Group 24 interrupt priority level register (LSB)
13	G24LV1	Group 24 interrupt priority level register
14	G24LV2	Group 24 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.22 Group 25 Interrupt Control Register

Register symbol: G25ICR

Address: x' 34000164

Purpose: This register is used to enable group 25 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G25 LV2	G25 LV1	G25 LV0	-	-	-	IQ2 IE	-	-	-	IQ2 IR	-	-	-	IQ2 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ2ID	External interrupt 2 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ2IR	External interrupt 2 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ2IE	External interrupt 2 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G25LV0	Group 25 interrupt priority level register (LSB)
13	G25LV1	Group 25 interrupt priority level register
14	G25LV2	Group 25 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.23 Group 26 Interrupt Control Register

Register symbol: G26ICR

Address: x' 34000168

Purpose: This register is used to enable group 26 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G26 LV2	G26 LV1	G26 LV0	-	-	-	IQ3 IE	-	-	-	IQ3 IR	-	-	-	IQ3 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ3ID	External interrupt 3 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ3IR	External interrupt 3 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ3IE	External interrupt 3 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G26LV0	Group 26 interrupt priority level register (LSB)
13	G26LV1	Group 26 interrupt priority level register
14	G26LV2	Group 26 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.24 Group 27 Interrupt Control Register

Register symbol: G27ICR

Address: x' 3400016C

Purpose: This register is used to enable group 27 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G27 LV2	G27 LV1	G27 LV0	-	-	-	IQ4 IE	-	-	-	IQ4 IR	-	-	-	IQ4 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ4ID	External interrupt 4 interrupt detection flag
		0: No interrupt detected
		1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ4IR	External interrupt 4 interrupt request flag
		0: No interrupt requested
		1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ4IE	External interrupt 4 interrupt enable flag
		0: Disabled
		1: Enabled
9 to 11	—	Always returns "0".
12	G27LV0	Group 27 interrupt priority level register (LSB)
13	G27LV1	Group 27 interrupt priority level register
14	G27LV2	Group 27 interrupt priority level register (MSB)
		Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.25 Group 28 Interrupt Control Register

Register symbol: G28ICR

Address: x' 34000170

Purpose: This register is used to enable group 28 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G28 LV2	G28 LV1	G28 LV0	-	-	-	IQ5 IE	-	-	-	IQ5 IR	-	-	-	IQ5 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ5ID	External interrupt 5 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ5IR	External interrupt 5 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ5IE	External interrupt 5 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G28LV0	Group 28 interrupt priority level register (LSB)
13	G28LV1	Group 28 interrupt priority level register
14	G28LV2	Group 28 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.26 Group 29 Interrupt Control Register

Register symbol: G29ICR

Address: x' 34000174

Purpose: This register is used to enable group 29 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G29 LV2	G29 LV1	G29 LV0	-	-	-	IQ6 IE	-	-	-	IQ6 IR	-	-	-	IQ6 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ6ID	External interrupt 6 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ6IR	External interrupt 6 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ6IE	External interrupt 6 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G29LV0	Group 29 interrupt priority level register (LSB)
13	G29LV1	Group 29 interrupt priority level register
14	G29LV2	Group 29 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.27 Group 30 Interrupt Control Register

Register symbol: G30ICR

Address: x' 34000178

Purpose: This register is used to enable group 30 interrupts, and to confirm interrupt requests and detection.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	G30 LV2	G30 LV1	G30 LV0	-	-	-	IQ7 IE	-	-	-	IQ7 IR	-	-	-	IQ7 ID
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit No.	Bit name	Description
0	IQ7ID	External interrupt 7 interrupt detection flag 0: No interrupt detected 1: Interrupt detected
1 to 3	—	Always returns "0".
4	IQ7IR	External interrupt 7 interrupt request flag 0: No interrupt requested 1: Interrupt requested
5 to 7	—	Always returns "0".
8	IQ7IE	External interrupt 7 interrupt enable flag 0: Disabled 1: Enabled
9 to 11	—	Always returns "0".
12	G30LV0	Group 30 interrupt priority level register (LSB)
13	G30LV1	Group 30 interrupt priority level register
14	G30LV2	Group 30 interrupt priority level register (MSB) Sets a level from 6 to 0.
15	—	Always returns "0".

7.4.2.28 Interrupt Acceptance Group Register

Register symbol: IAGR

Address: x' 34000200

Purpose: This register is used to read the number of a generated interrupt request.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	-	-	GN4	GN3	GN2	GN1	GN0	-	-
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Bit No.	Bit name	Description
0 to 1	—	Always returns "0".
2 to 6	GN0 to GN4	Group number register
		The group number that was accepted is stored in GN (4:0).
7 to 15	—	Always returns "0".

The Interrupt Accept Group Register (IAGR) stores the current lowest interrupt group number among the group numbers of the interrupts with the same interrupt level indicated by IM2 to IM0 in the PSW. Because the interrupt level is set to IM2 to IM0 when an interrupt is accepted by the CPU, IAGR indicates the current lowest interrupt group number among the group numbers of the interrupts that have been accepted by the CPU with the same interrupt level. If IM2 to IM0 is changed, or Group Interrupt Control Register is updated, or a new interrupt request occured, IAGR may change while the interupt handler is working.

The interrupt acceptance group register IAGR is a read-only register; it cannot be written.

7.4.2.29 External Interrupt Condition Specification Register

Register symbol: EXTMD

Address: x' 34000280

Purpose: This register specifies the external interrupt generation conditions. Set the desired level or edge for each pin.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	IR7 TG1	IR7 TG0	IR6 TG1	IR6 TG0	IR5 TG1	IR5 TG0	IR4 TG1	IR4 TG0	IR3 TG1	IR3 TG0	IR2 TG1	IR2 TG0	IR1 TG1	IR1 TG0	IR0 TG1	IR0 TG0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	IR0TG0	IRQ0 pin trigger condition setting (LSB)
1	IR0TG1	IRQ0 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
2	IR1TG0	IRQ1 pin trigger condition setting (LSB)
3	IR1TG1	IRQ1 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
4	IR2TG0	IRQ2 pin trigger condition setting (LSB)
5	IR2TG1	IRQ2 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
6	IR3TG0	IRQ3 pin trigger condition setting (LSB)
7	IR3TG1	IRQ3 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
8	IR4TG0	IRQ4 pin trigger condition setting (LSB)
9	IR4TG1	IRQ4 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
10	IR5TG0	IRQ5 pin trigger condition setting (LSB)
11	IR5TG1	IRQ5 pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge

<Continued>

<Continued>

Bit No.	Bit name	Description
12	IR6TG0	$\overline{\text{IRQ6}}$ pin trigger condition setting (LSB)
13	IR6TG1	$\overline{\text{IRQ6}}$ pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge
14	IR7TG0	$\overline{\text{IRQ7}}$ pin trigger condition setting (LSB)
15	IR7TG1	$\overline{\text{IRQ7}}$ pin trigger condition setting (MSB)
		00: Low level 01: High level
		10: Negative edge 11: Positive edge

Maintain the external pin interrupt signal value for at least four SYSCLK cycles when FRQS = 0, or for at least two SYSCLK cycles when FRQS = 1.

> **In stop mode, level detection is used, even if edge detection is set for the external interrupt detection condition. In other words, when positive edge detection is set, high-level detection results; when negative edge detection is set, low-level detection results.**

7.5   Description of Operation

The interrupt controller processes the following interrupts.

- Non-maskable interrupts
    $\overline{\text{NMIRQ}}$ pin interrupts
    Watchdog timer overflow interrupts
    System error interrupts

- Level interrupts

- Internal interrupts
    Peripheral interrupts from timers, serial I/O, A/D converter, watchdog timer, DMA

- External interrupts
    External pin interrupts: 8 pins
    (For external pin interrupt signals, maintain the value for at least four SYSCLK cycles when FRQS = 0, or for at least two SYSCLK cycles when FRQS = 1.)

In the event of a level interrupt, the interrupt group is determined and then an interrupt request is sent to the CPU.

If an interrupt signal is received, a determination is made as to whether it is a non-maskable interrupt or a level interrupt. If it is a level interrupt, the interrupt group is determined by deciding to which group the interrupt source belongs.

Once the interrupt group is determined, an interrupt request is generated by manipulating the interrupt control register (GnICR) corresponding to that group and notifying the CPU of the interrupt level. The interrupt group number is also set in the interrupt acceptance group register (IAGR).

The group interrupt level can be found by reading the interrupt priority level register LV (2:0) in the interrupt control register (GnICR).

If multiple level interrupt signals are received, the group to which each belongs is determined and then the interrupt group with the highest priority level is selected. When the group levels are the same, the group with the smallest group number is given priority.

For non-maskable interrupts, a non-maskable interrupt request is sent to the CPU without performing the processing described above.

1   Maintain the signal value for at least four SYSCLK cycles when FRQS = 0, or for at least two SYSCLK cycles when FRQS = 1 in the case of external pin interrupt request signals. Level detection is not possible if the signal is not maintained for at least the indicated number of clock cycles.

The above condition is not required in stop mode.

2   When writing a GnICR register in an interrupt program in order to clear IR and ID and then returning from the interrupt program, in order to gain synchronization with the bus controller store buffer be certain to perform an I/O bus access between the execution of the instruction (movhu, etc.) that is used to write the clear data to the GnICR register and the execution of the instruction to return from the interrupt program.

Example: After clearing a GnICR register, read it again.

```
mov     0x0f:b,d0     (d0= clear data)
movbu   d0,(GnICR)    Clears the GnICR flags.
                      (GnICR = address of GnICR register to be cleared)
movhu,  (GnICR),d1    I/O bus access
                      (Reads the GnICR register that was cleared)
rti                   Return from interrupt program
```

If there is no I/O bus access between the instruction that is used to write the clear data to the GnICR register and the instruction to return from the interrupt program, the return from the interrupt program is not guaranteed.

Particularly in a case such as that shown below where the return instruction follows immediately after the writing of the clear data, the interrupt program will be mistakenly executed again after the return.

```
mov     0x0f:b,d0
movbu   d0,(GnICR)    Clears the GnICR flags.
rti                   Return from interrupt program
```

3.   Before changing a GnICR register, clear the IE bits in the PSW.

Chapter 8.   8-bit Timers

8

8.1   Overview

The 8-bit timers have four built-in 8-bit reload timers, and can be used as interval timers or as event counters.

8.2   Features

• Clock source

Internal / external clock can be selected as clock source.

Internal clock: IOCLK, 1/8IOCLK, 1/32IOCLK, underflow of timer 0, 1, 2

External clock: count rise edge of input pins

• Cascaded connection

By cascading timer 0-3, they can be used as pure 16-, 24-, 32-bit timers.

• Interrupt

When timer underflow occurs, an interrupt is generated.

• Timer output

One half of cycle for timer under flow is possible as output.

• Generation of reference clock for serial interface. (timer 0,1,2,3)

• Generation of reference clock for UART. (timer 0,1,2,3)

• DMA can be started up when an interrupt request is generated. (timer 2,3)

8.3 System Configuration

8-bit Timer Connection Diagram
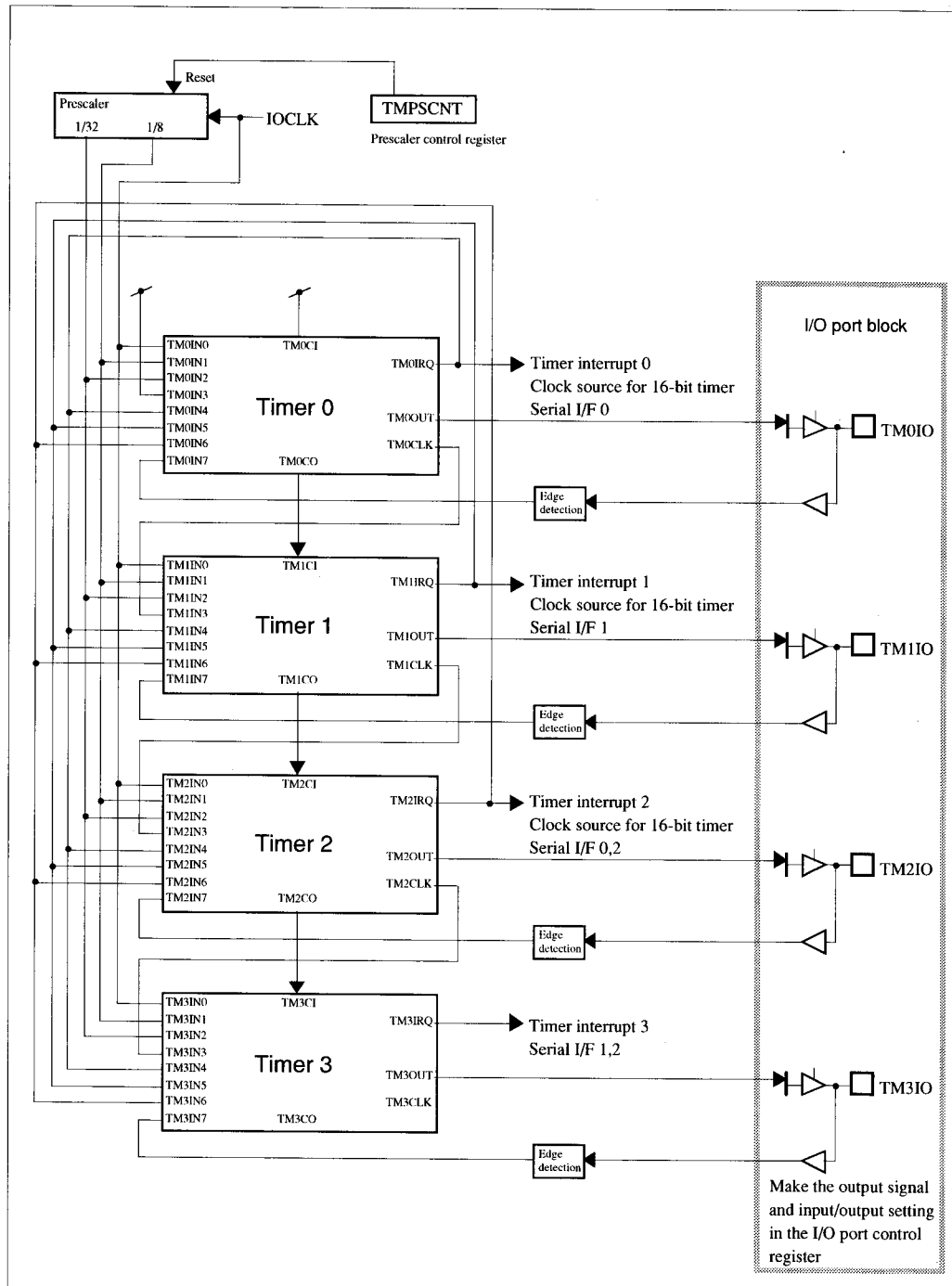
Fig. 8-3-1 Timer Connection Diagram

8.4 Block Diagram

8-bit Timer Block Diagram
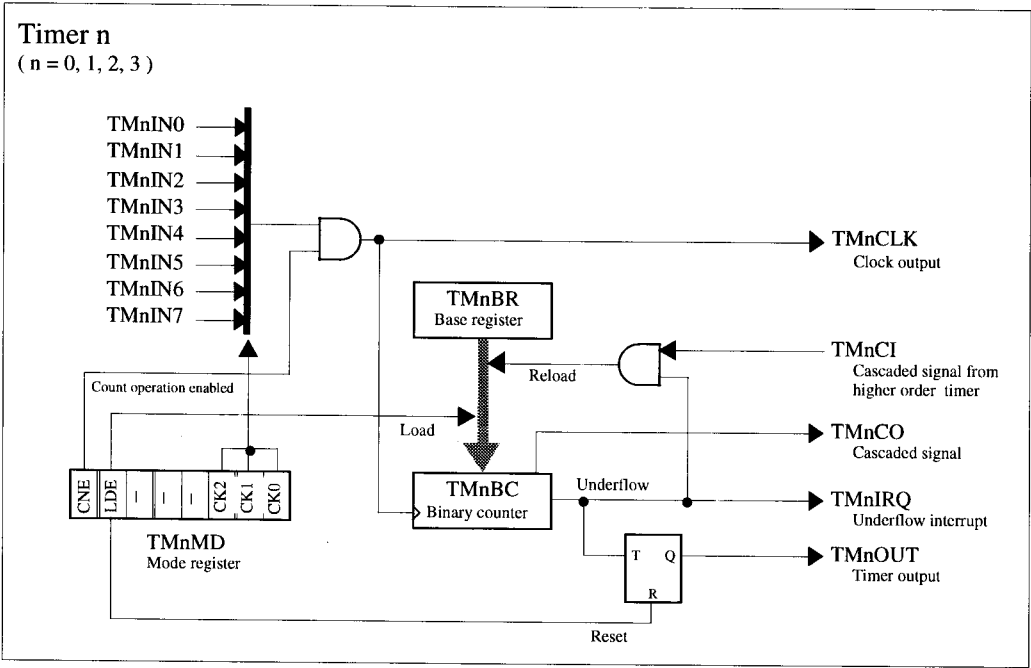
Fig. 8-4-1 Timer Configuration Diagram

8.5   List of Functions

8-bit Timer Function Chart

Table 8-5-1 Function Chart

	Timer 0	Timer 1	Timer 2	Timer 3
Interval timer	○	○	○	○
Event counter	○	○	○	○
Timer output	○	○	○	○
Interruput	○	○	○	○
Initiation source of DMA	×	×	○	○
16-bit timer source	○	○	○	×
Reference clock for serial I/F 0	○	×	○	×
Reference clock for serial I/F 1	×	○	×	○
Reference clock for serial I/F 2	×	×	○	○
Cascaded connection		○	○	○	

8.6 Description of Registers

Table 8-6-1 List of 8-bit Timer Registers

address	Name	Symbol	Number of bits	Initial value	Access size
x'34001000	Timer 0 mode register	TM0MD	8	x'00	8, 16, 32
x'34001001	Timer 1 mode register	TM1MD	8	x'00	8
x'34001002	Timer 2 mode register	TM2MD	8	x'00	8, 16
x'34001003	Timer 3 mode register	TM3MD	8	x'00	8
x'34001010	Timer 0 base register	TM0BR	8	x'00	8, 16, 32
x'34001011	Timer 1 base register	TM1BR	8	x'00	8
x'34001012	Timer 2 base register	TM2BR	8	x'00	8, 16
x'34001013	Timer 3 base register	TM3BR	8	x'00	8
x'34001020	Timer 0 bainary counter	TM0BC	8	x'00	8, 16, 32
x'34001021	Timer 1 bainary counter	TM1BC	8	x'00	8
x'34001022	Timer 2 bainary counter	TM2BC	8	x'00	8, 16
x'34001023	Timer 3 bainary counter	TM3BC	8	x'00	8
x'34001071	Prescaler control register	TMPSCNT	8	x'00	8

Prescaler timer is also used in 16-bit timer.

When writing to or reading from an 8-bit timer register, set the I/O bus mode to synchronous mode. Operation is not guaranteed if the read/write is performed in asynchronous mode.

8.6.1　Timer n Mode Register (n = 0, 1, 2, 3)

Register symbol:　　　TMnMD

Address:　　　　　　x'34001000 (n=0) , x'34001001 (n=1)

　　　　　　　　　　x'34001002 (n=2) , x'34001003 (n=3)

Purpose:　　　　　　Control of the operation of timet n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CNE	TMn LDE	-	-	-	TMn CK2	TMn CK1	TMn CK0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMnCK0	Clock source selection (LSB)
1	TMnCK1	Clock source selection
2	TMnCK2	Clock source selection (MSB)
		Refer to table 8-6-2 about clock source setting
3 to 5	—	Always returns "0"
6	TMnLDE	Timer n initialization flag
		Initializes timer n.
		0: Normal operation
		1: Initialize
		Loads the value of TMnBR in TMnBC. Resets timer output n to "L" level.
7	TMnCNE	Timer n output enable flag
		Enables/disables the timer n count operation.
		0: Operation disabled
		1: Operation enabled

To set TMnCK0, 1, 2, TMnCNE has to be set 0.

To set TMnLDE equal to 0, TMnCNE has to be set 1.

To set TMnCNE equal to 0, TMnLDE has to be set 1.

Operation is not guaranteed , when both TMnCNE and TMnLDE are become 1.

Table 8-6-2 8-bit Timer Clock Source

TMnCK[2:0] Setting	Timer 0	Timer 1	Timer 2	Timer 3
000	IOCLK	IOCLK	IOCLK	IOCLK
001	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK
010	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK
011	Setting prohibited	Cascaded with timer 0	Cascaded with timer 1	Cascaded with timer 2
100	Setting prohibited	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Setting prohibited	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Setting prohibited	Timer 2 underflow
111	TM0IO pin input	TM1IO pin input	TM2IO pin input	TM3IO pin input

You must need to set prescaler control register (TMPSCNT), when 1/ 8 IOCLK or 1/32 IOCLK is used. If TMnIO pin input is selected, count by rise edge of pin input.

8.6.2　Timer n Base Register (n = 0, 1, 2, 3)

Register symbol:　　TMnBR

Address:　　　　　　x'34001010 (n=0) , x'34001011 (n=1)

　　　　　　　　　　x'34001012 (n=2) , x'34001013 (n=3)

Purpose:　　　　　　Set the initial value of the timer n binary counter and the underflow cycle.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn BR7	TMn BR6	TMn BR5	TMn BR4	TMn BR3	TMn BR2	TMn BR1	TMn BR0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Under below condition, the setting value of TMnBR is loaded to TMnBC.

　　(1) When TMnLDE=1.

　　(2) When timer underflow is occurred.

Whenever TMnBC counts the value of (resisted value of TMnBR + 1), TMnBC become undeflow, then occurs the request of underflow interrupt.

8.6.3　Timer n Binary Counter　(n = 0, 1, 2, 3)

Register symbol:　　　TMnBC

Address:　　　　　　x'34001020 (n=0) , x'34001021 (n=1)

　　　　　　　　　　x'34001022 (n=2) , x'34001023 (n=3)

Purpose:　　　　　　This register is the binary counter (down-counter) for timer n.

　　　　　　　　　　The counter value can be read from this register.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn BC7	TMn BC6	TMn BC5	TMn BC4	TMn BC3	TMn BC2	TMn BC1	TMn BC0
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

It is down counter.

Whenever TMnBC counts the value of (the value of TMnBR + 1), TMnBC become undeflow, then occurs the request of underflow interrupt.

8.6.4　Prescaler Control Register

Register symbol:　　　TMPSCNT

Address:　　　　　　x'34001071

Purpose:　　　　　　This register controls the prescaler opreation.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMPS CNE	—	—	—	—	—	—	—
When reset	0	0	0	0	0	0	0	0
Access	R/W	R	R	R	R	R	R	R

Bit No.	Bit name	Description
0 to 6	—	Always returns "0".
7	TMPSCNE	Prescaler operation enable flag.
		Enables/disables operation of the 1/8 IOCLK and 1/32 IOCLK prescaler.
		0: Operation disabled
		1: Operation enabled

8.7 Description of Operation

8.7.1 Interval Timer, Timer Output

While use 8-bit timer as interval timer, set as follows.

Then operate as interval timer which occurs the interrupt signal with every setting cycle.

(refer to figure8-7-1,figure8-7-2, figure8-7-3)

If timers are cascaded, and operate as 16-,24-,32-bit timer, refer to the chapter 8.7.3 "Cascaded Connection".

Sequence of Startup Operation

(1) Set the division rate of timer

Set the division rate to TMnBR.

The cycle time for interrupt becomes "(value of TMnBR+1) x cycle time of clock source".

(2) Select clock source

Clock source can be selected by TMnCK[2:0] fields of TMnMD register.

If use 1/8 IOCLK, 1/32 IOCLK as clock source, enable prescaler operation by setting vale 1 to TMPSCNE field of TMPSCNT register.

(3) Initiate timer

Initiate the timer n setting vale 1 to TMnLED field of TMnMD register.

Value of TMnBR is loaded to TMnBC as initial data, then reset timer output.

After initiate time, set certainly normal mode by setting value 0 to TMnLED register.

(4) Set I/O ports (if use timer output)

Set I/O port to timer output pins by setting I/O port register .

Refer to the chapter "I/O port" how to set I/O port register.

(5) Permit timer counting operation

Start timer counting operation , if set value 1 to TMnCNE field of TMnMD register.

If permit timer counting operation, generate the request of underflow interrupt with specified cycle.

Whenever interrupt occurs, condition of output pin is negated, then load the value of TMnBR to TMnBC.

If change the value of TMnBR during counting operation, when next underflow occurs, that is used as initial value, and interrupt cycle will be changed.

Sequence of Stopping Operation

(1) Stop timer counting

Stop timer counting, when set vale 0 to TMnCNE field of TMnMD register.

(2) Initialize timer if it is necessary.

If set value 1 to TMnLED field of TMnMD register, the value of TMnBR register is loaded into TMnBC as initial value, then output is reset.

After timer is stopped, if never set the value 1 to TMnLDE, the value output from pin and binary counter are maintained.

If set the value 1 to TMnCNE again, restart counting from the last stopping condition.
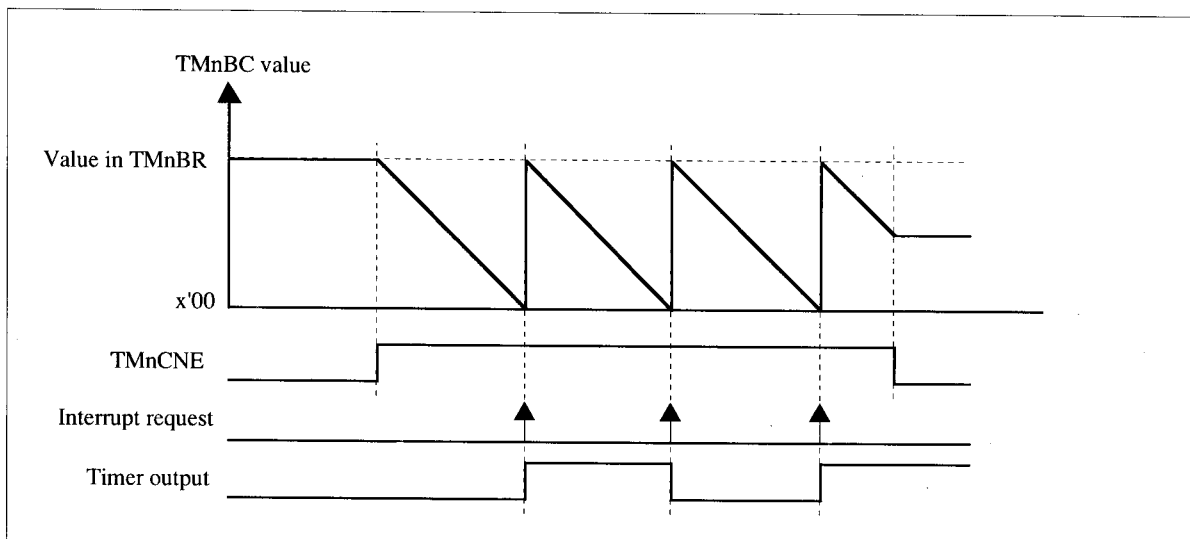
Fig. 8-7-1 Interval Timer Operation
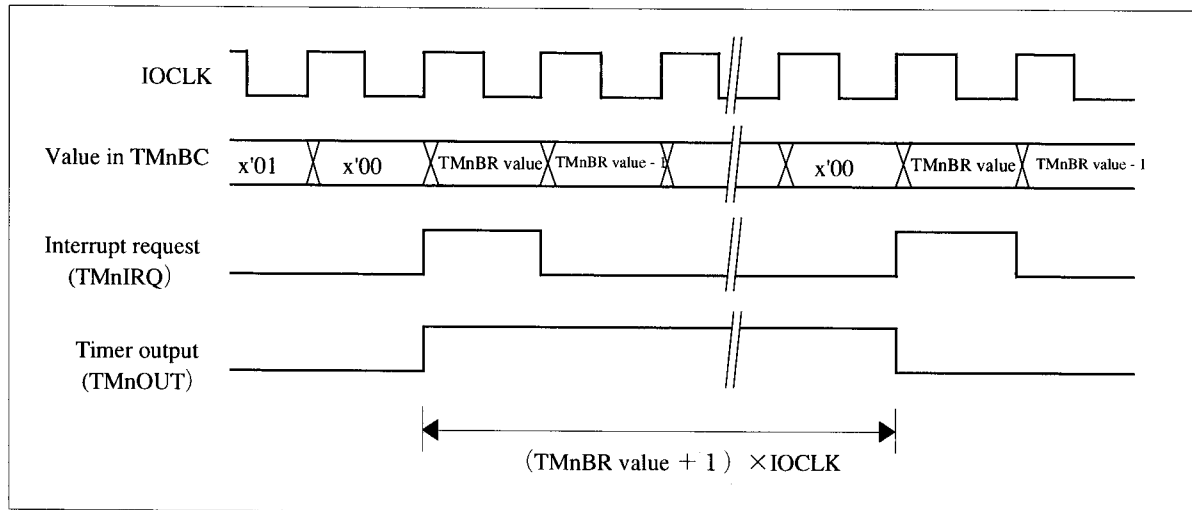
Fig. 8-7-2 Interval Timer Operation (When Clock Source = IOCLK)
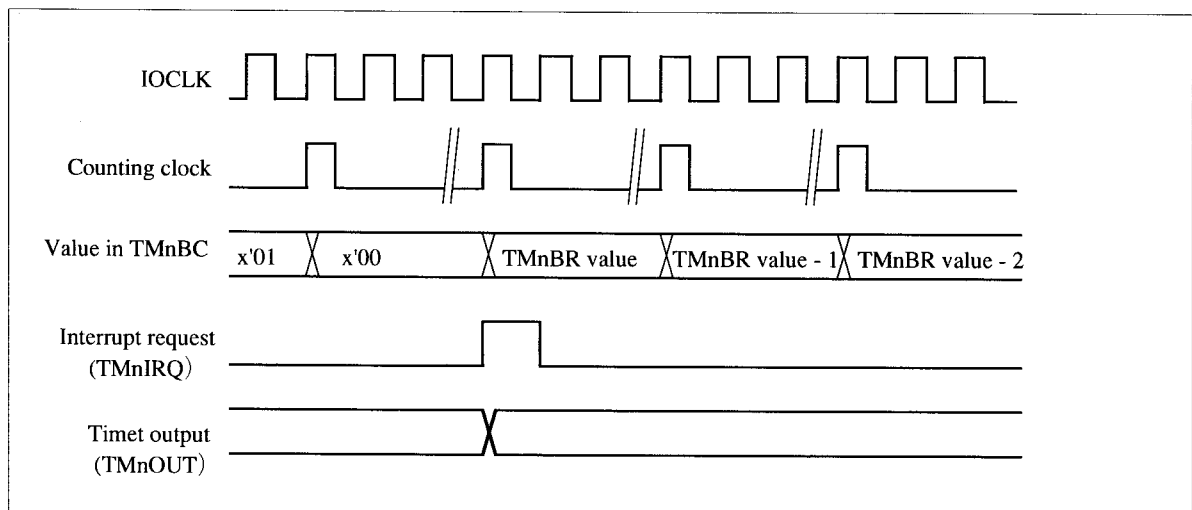
Fig. 8-7-3 Interval Timer Operation (When Use Prescaler)

8.7.2 Event Counting Operation

While 8-bit timer is used as event counting, set as follows.
If timers are cascaded, and operate as 16-,24-,32-bit timer, refer to the chapter 8.7.3 "Cascaded Connection".

Sequence of Startup Operation

    (1) Set the division ratio of timer

        Set the division ratio to TMnBR.

        The request of interrupt occurs, after count (value of TMnBR+1) times rise edge of input.

    (2) Select Clock Source

        Set clock source to TMnIO pin by handling TMnCK[2:0] field of TMnMD register.

    (3) Initialize Timer

        Initialize timer n by setting the value 1 to TMnLDE field of TMnMD register.

        Then the value of TMnBR register is loaded into TMnBC register as initial value.

        After initialization, return to the normal mode by setting the value 0 to TMnLDE.

    (4) Set I/O port

        Set I/O port as input pin.

        Refer to the chapter "I/O port" about setting registers.

    (5) Permit timer counting operation

        Permit timer counting by setting the value 1 to TMnCNE field of TMnMD register.

Once permit timer counting, the rise edge of the port input are counted, then generates interrupt signal on underflow in binary counter, and load the value of TMnBR register into TMnBC register.
If change the value of TMnBR register while counting in progress, that is used as initial value when next underflow occurs.

Sequence of Stopping Operation

    (1) Stop timer counting

        Stop timer counting, when set the value 0 to TMnCNE field of TMnMD register.

    (2) Initialize timer if it is necessary.

        If set value 1 to TMnLED field of TMnMD register, the value of TMnBR register timer is loaded into TMnBC as initial value.

        After timer is stopped, if never set the value 1 to TMnLDE, the value of binary counter are maintained the last modified value.

        If set the value 1 to TMnCNE again, restart counting from the last stopping condition.

> ⚠ **Input pulse width must be more than IOCLK x 1.5 since IOCLK being sampled at pin input.**
> **Event count operation is inhibited when IOCLK is terminated (HALT, STOP mode).**
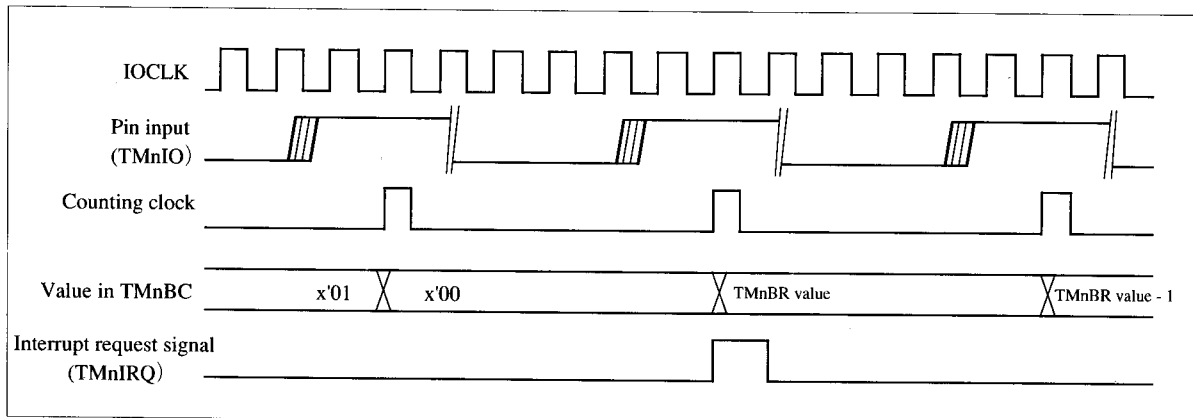
Fig. 8-7-4 Event Counting Operation

8.7.3 Cascaded Connection
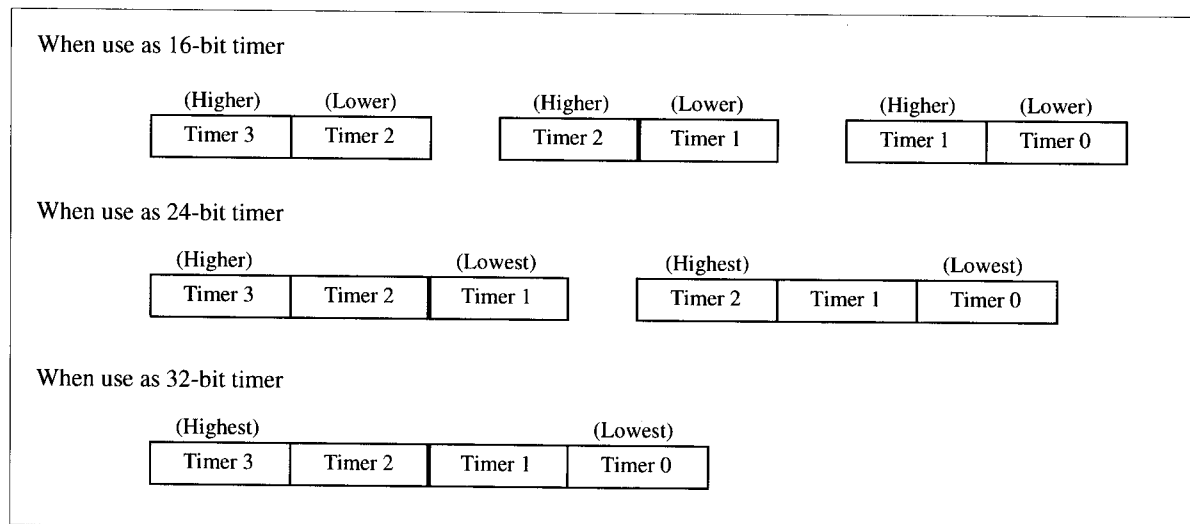
Fig. 8-7-5 Cascaded Connection

If use as cascaded timers, set as follows.

(1) Set the division ratio of the timer

(Example 1) If the cascaded timer 0, 1 are cascaded(16-bit timer), it is necessary to set x'1233

(x'1234 -1) to TMnBR register that division ratio is set to x'1234.

Then set x'33 to TM0DR, x'12 to TM1BR.

TMnBR register can be accessed with 16-/ 32-bit width, so set the value to the plural TMnBR register by using one instruction.

(If timer 1,2 are cascaded as 16-bit timer or 3 timers are cascaded as 24-bit timer, never set to these TMnBR by using one instruction.)

If change the value of TMnBR on counting operation, change TMnBR registers simultaneously which are cascaded timers.

(2) Select clock source

Select clock source as the lowest order timer.

The clock sources of upper order timers are set to "cascaded connection".

(example 1)   If timer 0,1 are cascaded as 16-bit timer:

set the clock source of timer 0 .

set the clock source of timer 1to "cascaded connection".

(example 2)   If timer 0,1,2,3 are cascaded as 32-bit timer.

set the clock source of timer 0 .

set the clock source of timer 1,2,3 to "cascaded connection".

(3) Initialize timer

Set all TMnLDE flag in which timers are cascaded, then initialize timers.

(no need to set the value to registers simultaneously.)

(4) Permit counting operation

Perform one of which listed below due to permit timer counting.

1) Start to permit counting operation from upper timer which is cascaded.

2) Permit counting all timer which is cascaded simultaneously.

(5) Stop counting operation

Perform one of which listed below due to stop timer counting.

1)Start to stop counting operation from lower timer which is cascaded.

2) Stop counting all timer which is cascaded simultaneously.

(6) Timer output, interrupt

Only the timer output or the interrupt request from the highest timer which is cascaded is available.

The timer output or the interrupt request from the lower timer is out of warranty.

8.7.4 Example of Prescaler Timer and Cascaded Timer

(1) When set clock source of timer 1 to "underflow flag of timer 0"

After TM0BC become underflow, the value of TM0BR is loaded into TM0BC, and the value of TM1BC decrement.

After TM1BC become underflow, the value of TM1BR is loaded into TM1BC.
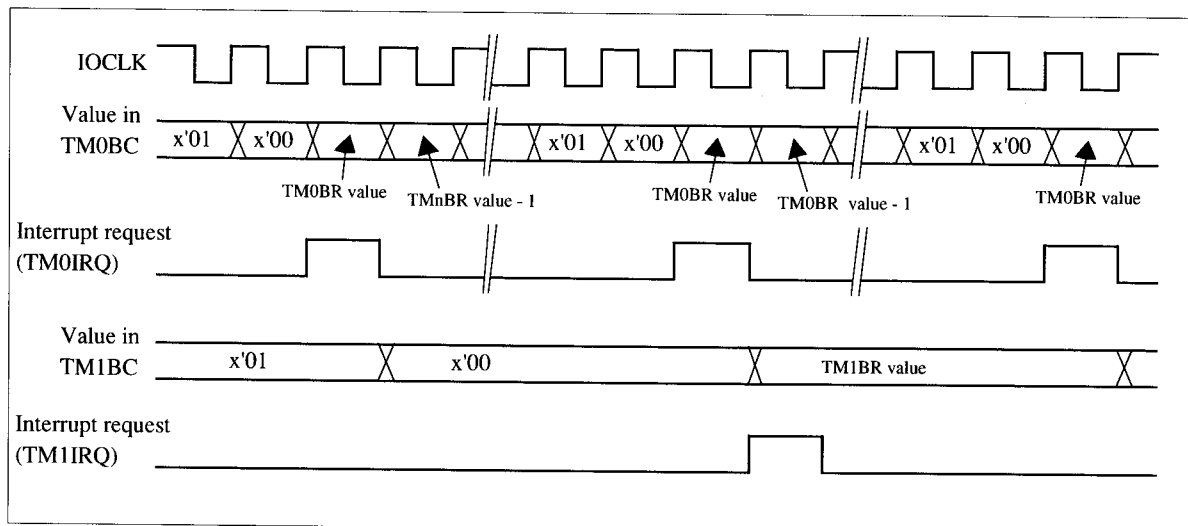
Fig. 8-7-6 Operation of Timer 0 and 1 (1)

(2)When timer0,1 are cascaded

During TM1BC ≠ x'00, when TM0BC become underflow, TM0BC is set to x'FF, and the value of TM1BC decrement.

During TM1BC = x'00, when TM0BC become underflow, the value of TMoBR,TM1BR are loaded into TM0BC, TM1BC, and the interrupt request of timer 1 occurs.
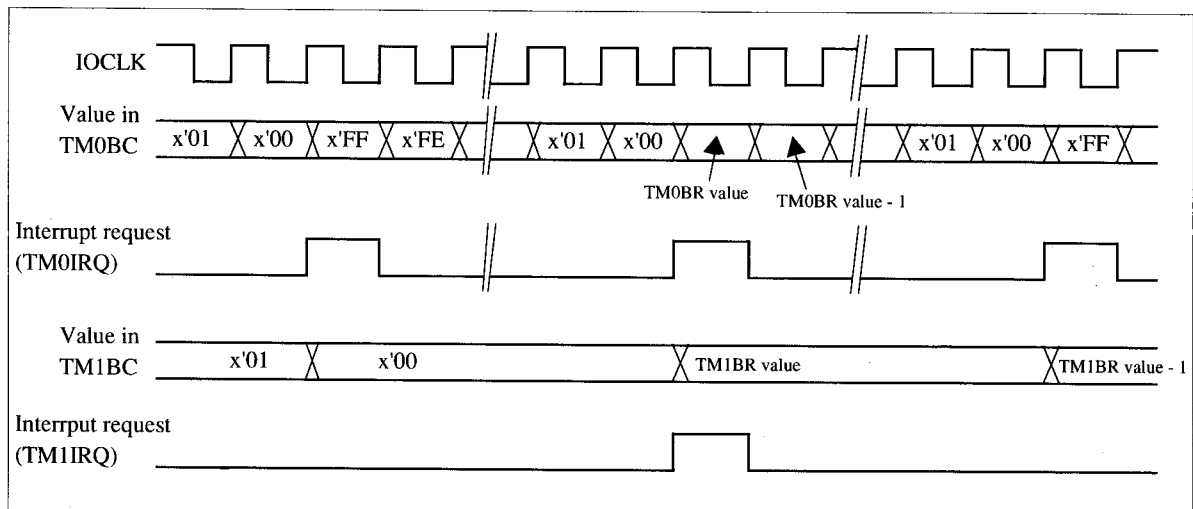
Fig. 8-7-7 Operation of Timer 0 and 1 (2)

Chapter 9.   16-bit Timers

9

9.1 Overview

The 16-bit timers include three built-in 16-bit timer:

Two of them are reload timers (down counter), and can be used as either of interval timer or events counter.

The other one is up counter and has internal two compare/capture registers.

9.2 Features

Timer 4, 5

- Reload timer (Down counter)

- Clock source
    Available to use either of internal clock or external clock as clock source.
    Internal clock: IOCLK, 1/8 IOCLK, 1/32 IOCLK, Under flow of timer 0-2 are available.
    External clock: Count the rising edge of the pin input.

- Cascaded connection
    By cascading timer 4,5, they can be used as pure 32-bit timer.

- Interrupts
    Generate interrupt when a timer underflow occurs.

- Timer output
    One half of cycle for timer underflow is possible as output.

Timer 6

- Up counter

- Clock source
    Internal/External clock are available for clock source.
    Internal clock: IOCLK, 1/8 IOCLK, 1/32 IOCLK, Under flow of timer 0-2 are available.
    External clock: count either of rising edge or falling edge of TM6IOB pin input.

- Compare/capture register
    Has two compare/capture register built in.

• Pin output

Capable of PWM output with variable cycle and duty ratio. (One signal)

Capable of PWN output with added bit. (Resolution : 8+2 bits, 8+3 bits, 8+4 bits, 8+6 bits)

Capable of one shot output. (Two signals)

Output polarity setting possible.

• Input capture

For each pin, it is available to set to one of following: Rise edge Fall edge or both edge. (Two signals)

Interrupts request occurs upon capture.

If it's set to both rising and falling edge, interrupts occur at both rising and fall edge.

• Interrupts

Occur when binary counter overflows.

Occur when compare register and binary counter matches or when capture occur. (Two signals)

• Counter start due to external trigger

Counter starts with the TMn6IOB pin input. (Rising edge/falling edge)

• DMA request

Occur when binary counter and compare/capture register matches.

(For the case of additional bit style PWM output mode, DMA request occurs when binary counter overflow happens.)

9.3　System Configuration
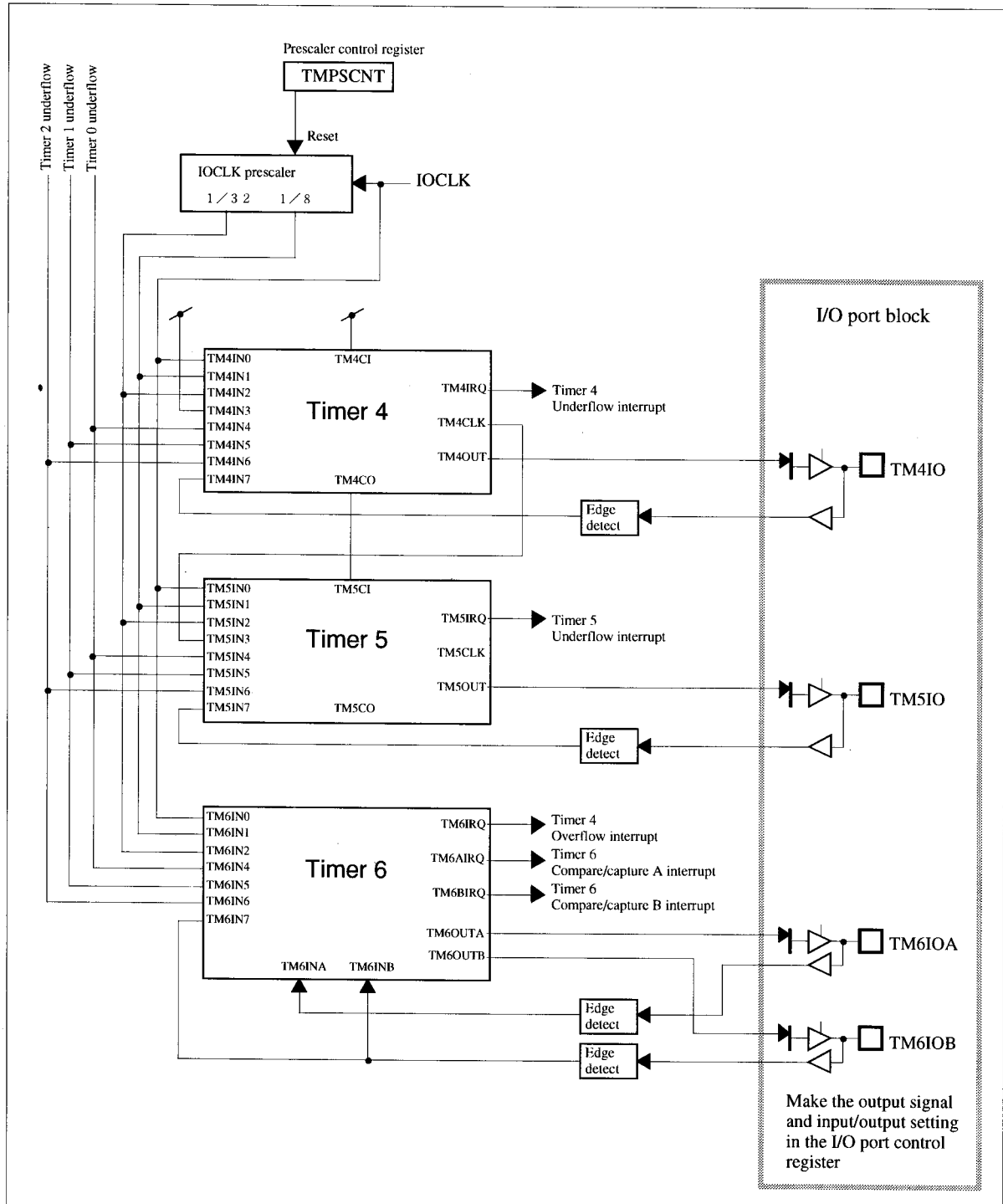
Fig. 9-3-1　16-bit Timer Connection Diagram

9.4 Block Diagram

Timer 4 (Timer 5)

Fig. 9-4-1 16-bit Timer Block Diagram (Timer 4, 5)

Timer 6

Fig. 9-4-2 16-bit Timer Block Diagram (Timer 6)

Timer 6 compare/capture registers (TM6CA, TM6CB)
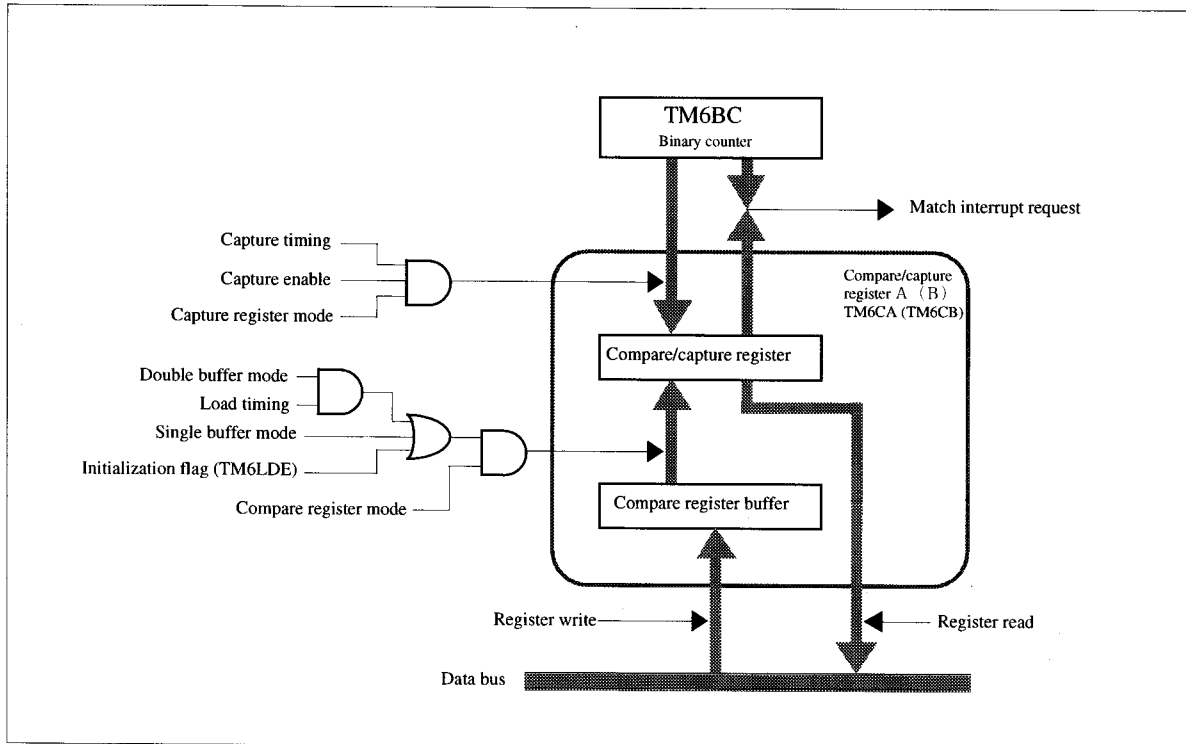
Fig. 9-4-3 Timer 6 Compare/Capture Register Block Diagram

PWM output section when timer 6 is set to PWM mode with additional bits
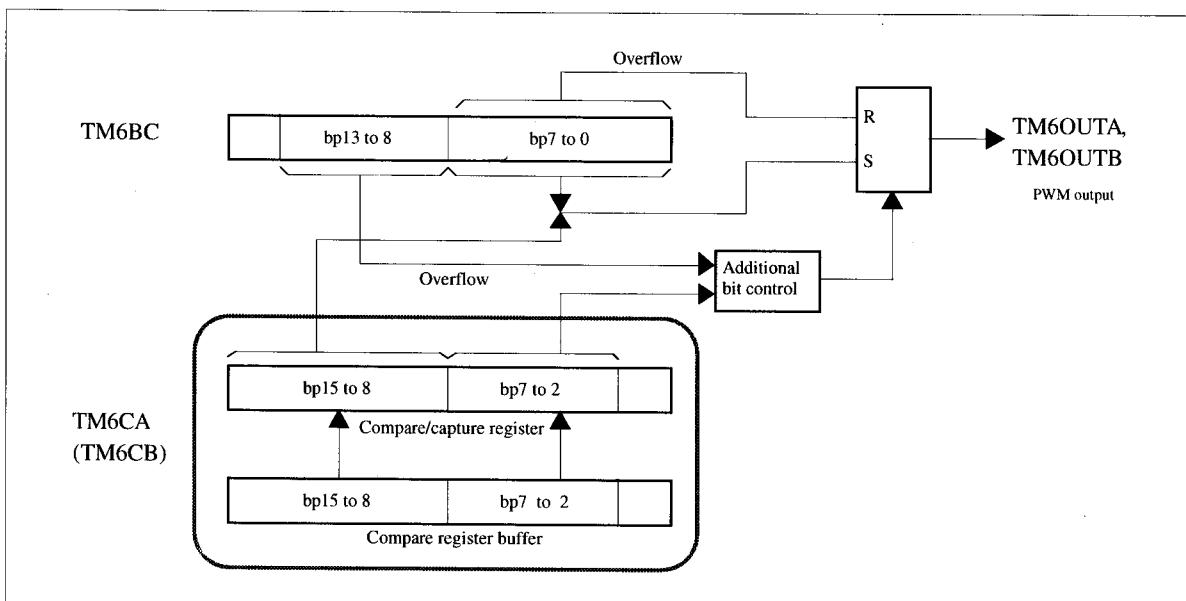
Fig. 9-4-4 PWM Output Section Block Diagram

9.5   List of Functions

Table 9-5-1 16-bit Timer Functions

	Timer 4	Timer 5	Timer 6
Up/down counter	Down counter	Down counter	Up counter
Inerval timer	○	○	○
Event counting	○ (Rising edge)	○ (Falling edge)	○ (Dual edge)
Number of compare/ capture register	—	—	2
Toggled output	○	○	○ (2)
PWM output	×	×	Variable cycle, duty ...1 Fixed cycle ...2
Interrupts	Underflow	Underflow	Overflow Compare/capture A Compare/capture B
Input capture	×	×	2 (Either single edge or both edges)
One-shot output	×	×	○
Initiation source of DMA	×	×	Compare/capture A
Cascaded connection		○	×	

9.6 Description of Registers

Table 9-6-1 List of 16-bit Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
0x34001080	Timer 4 mode register	TM4MD	8	x'00	8, 32
0x34001082	Timer 5 mode register	TM5MD	8	x'00	8
0x34001084	Timer 6 mode register	TM6MD	1 6	x'0000	8, 16
0x34001090	Timer 4 base register	TM4BR	1 6	x'0000	16
0x34001092	Timer 5 base register	TM5BR	1 6	x'0000	16
0x340010A0	Timer 4 binary counter	TM4BC	1 6	x'0000	16
0x340010A2	Timer 5 binary counter	TM5BC	1 6	x'0000	16
0x340010A4	Timer 6 binary counter	TM6BC	1 6	x'0000	16
0x340010B4	Timer 6 compare/capture A mode register	TM6MDA	8	x'00	8, 16
0x340010B5	Timer 6 compare/capture B mode register	TM6MDB	8	x'00	8
0x340010C4	Timer 6 compare/capture A register	TM6CA	1 6	x'0000	16
0x340010D4	Timer 6 compare/capture B register	TM6CB	1 6	x'0000	16
0x34001071	Prescaler control register	TMPSCNT	8	x'00	8

Prescaler control register (TMPSCNT) is also used at 8-bit timer.

Set I/O bus to synchronous mode for read/write of 16-bit timer.

Operation is out of guarantee for read/write action under asynchronous mode.

9.6.1 Timer n Mode Register (n = 4, 5)

Register symbol: TMnMD

Address: x'34001080 (n=4), x'34001082 (n=5)

Purpose: This register sets the operation control conditions for timer n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMn CNE	TMn LDE	—	—	—	TMn CK2	TMn CK1	TMn CK0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TMnCK0	Clock source selection (LSB)
1	TMnCK1	Clock source selection
2	TMnCK2	Clock source selection (MSB)
		Refer to table 9-6-2 about clock source setting
3 to 5	—	Always returns "0".
6	TMnLDE	Timer n initialization flag
		Initializes timer n.
		0: Normal operation
		1: Initialize
		Loads the value in TMnBR into TMnBC, and resets timer output n.
7	TMnCNE	Timer n operation enable flag
		Enables/disables the timer n counting operation.
		0: Operation stopped
		1: Operation enabled

> **To set TMnCK0, 1, 2, TMnCNE has to be set 0.**
>
> **To set TMnLDE equal to 0, TMnCNE has to be set 1.**
>
> **To set TMnCNE equal to 0, TMnLDE has to be set 1.**
>
> **Operation is not guaranteed, when both TMnCNE and TMnLDE are become 1.**

Table 9-6-2 16-bit Timer Clock Source

TMnCK [2:0] Setting	Timer 4	Timer 5	Timer 6
000	IOCLK	IOCLK	IOCLK
001	1/8 IOCLK	1/8 IOCLK	1/8 IOCLK
010	1/32 IOCLK	1/32 IOCLK	1/32 IOCLK
011	Setting prohibited	Cascaded with timer 4	Setting prohibited
100	Timer 0 underflow	Timer 0 underflow	Timer 0 underflow
101	Timer 1 underflow	Timer 1 underflow	Timer 1 underflow
110	Timer 2 underflow	Timer 2 underflow	Timer 2 underflow
111	TM4IO pin input	TM5IO pin input	TM6IOB pin input

To use 1/8 IOCLK, 1/32IOCLK, it is necessary to specify the setting of the prescaler register, TMPSCNT.

For timer 4 and 5, count at the rising edge of pin input if TMnIO pin input is selected.

For timer 6, count according to the TM6MDB register setting if TM6IOB pin input is selected.

9.6.2　Timer 6 Mode Register

Register symbol:　　TM6MD

Address:　　x'34001084

Purpose:　　This register sets the operation control conditions for timer 6.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM6 CNE	TM6 LDE	TM6 PME	TM6 PM1	TM6 PM0	—	—	—	TM6 TGE	TM6 ONE	—	TM6 CAE	—	TM6 CK2	TM6 CK1	TM6 CK0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM6CK0	Clock source selection (LSB)
1	TM6CK1	Clock source selection
2	TM6CK2	Clock source selection (MSB)
		Refer to Table 9-6-2 about clock source setting.
3	—	Always returns "0".

<Continued>

<Continued>

Bit No.	Bit name	Description
4	TM6CAE	Counter clear enable flag.

Enables/disables clearing of TM6BC when TM6BC and TM6CA match.

    0: Do not clear. (TM6BC becomes a 16-bit free-running counter.)

    1: Clear.

        When TM6CA is set as a compare register.

            TM6BC is cleared if TM6BC and TM6CA match.

        When TM6CA is set as a capture register.

            TM6BC is cleared when a capture occurs in TM6CA.

| 5 | — | Always returns "0". |
| 6 | TM6ONE | One-shot operation enable flag |

Enables/disables the halt of timer operation when TM6BC and TM6CA match.

    0: Disables one-shot operation.

    1: Enables one-shot operation .

        If TM6BC and TM6CA match, the TM6CNE flag is reset and the timer stops.

| 7 | TM6TGE | External trigger start enable flag |

Enables/disables timer start by an external trigger.

    0: Disables timer start by an external trigger.

        (The trigger input is ignored.)

    1: Enables timer start by an external trigger.

        When the specified edge is input to the TM6IOB pin, the TM6CNE flag is set and the timer starts.

        The timer starts on the edge that is the opposite of the one selected by the TM6BEG flag in the TM6MDB register.

8 to 10	—	Always returns "0".
11	TM6PM0	Resolution selection (LSB)
12	TM6PM1	Resolution selection (MSB)

Valid only in additional bit style PWM mode

    00: 10 bits (basic output: 8 bits; additional bits: 2 bits)

    01: 11 bits (basic output: 8 bits; additional bits: 3 bits)

    10: 12 bits (basic output: 8 bits; additional bits: 4 bits)

    11: 14 bits (basic output: 8 bits; additional bits: 6 bits)

<Continued>

<Continued>

Bit No.	Bit name	Description
13	TM6PME	Timer 6 PWM output waveform selection flag
		This bit selects the PWM output waveform for timer 6.
		0: Normal waveform
		1: PWM output with additional bits
		The PWM waveform is output with the resolution that is set in TM6PM1 and 0.
		The number of bits that was set in TM6PM1 and 0 is the number of bits in TM6BC that function as a binary counter.
14	TM6LDE	Timer 6 initialization flag
		Initializes timer 6.
		0: Normal operation
		1: Initialize
		Clears TM6BC (so that TM6BC = x'0000).
		When TM6CA and TM6CB are set as a double-buffer compare register, the value in the buffers is loaded into the compare register.
		Also initializes the pin output.
15	TM6CNE	Timer 6 enable flag
		Enable/disable the timer 6 counting operation
		0: Operation stopped.
		1: Operation enabled.

To set TM6CK0, 1, 2, TM6CNE has to be set 0.

To set TM6LDE equal to 0, TM6CNE has to be set 1.

To set TM6CNE equal to 0, TM6LDE has to be set 1.

Operation is not guaranteed, when both TM6CNE and TM6LDE are become 1.

9.6.3　Timer n Base Register (n = 4, 5)

Register symbol:　　　TMnBR

Address:　　　　　　　x'34001090 (n=4), x'34001092 (n=5)

Purpose:　　　　　　　This register sets the count cycle for timer n.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TMn BR15	TMn BR14	TMn BR13	TMn BR12	TMn BR11	TMn BR10	TMn BR9	TMn BR8	TMn BR7	TMn BR6	TMn BR5	TMn BR4	TMn BR3	TMn BR2	TMn BR1	TMn BR0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Under the following condition the value in TMnBR is loaded to TMnBC.

(1) TMnLDE = 1

(2) When underflow occurs

Underflow interrupts are generated whenever TMnBC counts (TMnBR value +1) times.

9.6.4　Timer n Binary Counter (n = 4, 5, 6)

Register symbol:　　　TMnBC

Address:　　　　　　　x'340010A0 (n=4), x'340010A2 (n=5)

　　　　　　　　　　　x'340010A4 (n=6)

Purpose:　　　　　　　This register is the binary counter (down-counter) for timer n.

　　　　　　　　　　　The counter value can be read from this register.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TMn BC15	TMn BC14	TMn BC13	TMn BC12	TMn BC11	TMn BC10	TMn BC9	TMn BC8	TMn BC7	TMn BC6	TMn BC5	TMn BC4	TMn BC3	TMn BC2	TMn BC1	TMn BC0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Timer 4 and 5 are down-counters, having TMnBR value as the initial value.

They generate interrupt when they make underflow caused on (TMnBR value +1) times count.

Timer 6 is a up-counter start from the initial value, x'0000.

It generates interrupt whenever overflow occurs.

Under additional bit style PWM mode, it operates as binary counter, and generates interrupt when overflow occurs.

9.6.5 Timer 6 Compare/Capture A Mode Register

Register symbol: TM6MDA

Address: x'340010B4

Purpose: This register sets the operation control conditions for the compare/capture register A for timer 6.

This register also sets the waveform that is output to the TM6IOA pin.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TM6 AM1	TM6 AM0	TM6 AEG	TM6 ACE	–	TM6 AO2	TM6 AO1	TM6 AO0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM6AO0	Timer 6A output waveform selection flag (LSB)
1	TM6AO1	Timer 6A output waveform selection flag
2	TM6AO2	Timer 6A output waveform selection flag (MSB)

These bits select the waveform that is output to the TM6IOA pin.

    000: Set when TM6BC matchs TM6CA, reset when TM6BC matchs TM6CB.

    001: Set when TM6BC matchs TM6CA, reset when TM6BC overflows.

    010: Set when TM6BC matchs TM6CA.

        (Reset only when the timer is initialized)

    011: Reset when TM6BC matchs TM6CA.

    100: Toggled output (Output is inverted when TMnBC matches TMnCA)

    101, 110, 111: Setting prohibited.

| 3 | — | Always returns "0". |
| 4 | TM6ACE | Timer 6 capture A operation enable flag |

Enables/disables caputure operation for TM6CA

    0:    Disables caputure operation. (Pin input is ignored.)

    1:    Enables caputure operation.

| 5 | TM6AEG | Timer 6 A pin polarity selection flag |

Selects the valid edge for the input on the TM6IOA pin, and the output polarity.

    0:    Rising edge valid

        Positive polarity output (Reset: "L" level; set: "H" level)

    1:    Falling edge valid

        Negative polarity output (Reset: "H" level; set: "L" level)

| 6 | TM6AM0 | Timer 6 compare/capture A operating mode setting flag (LSB) |
| 7 | TM6AM1 | Timer 6 compare/capture A operating mode setting flag (MSB) |

These bits set the TM6CA operating mode

    00:    Compare register (single-buffer)

    01:    Compare register (double-buffer)

    10:    Capture register (single-edge operation)

    11:    Capture register (dual-edge operation)

When dual-edge capture mode is set, the setting of TM6AEG is ignored.

9.6.6　Timer 6 Compare/Capture B Mode Register

Register symbol:　　TM6MDB

Address:　　　　　x'340010B5

Purpose:　　　　　This register sets the operation control conditions for the compare/capture register B for timer 6.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TM6 BM1	TM6 BM0	TM6 BEG	TM6 BCE	-	TM6 BO2	TM6 BO1	TM6 BO0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	TM6BO0	Timer 6B output waveform selection flag (LSB)
1	TM6BO1	Timer 6B output waveform selection flag
2	TM6BO2	Timer 6B output waveform selection flag (MSB)

These bits select the waveform that is output to the TM6IOB pin.

000: Set when TM6BC matchs TM6CB, reset when TM6BC matchs TM6CA.

001: Set when TM6BC matchs TM6CB, reset when TM6BC overflows.

010: Set when TM6BC matchs TM6CB.

(Reset only when the timer is initialized)

011: Reset when TM6BC matchs TM6CB.

100: Toggled output (Output is inverted when TMnBC matches TMnCB)

101, 110, 111: Setting prohibited.

| 3 | — | Always returns "0". |
| 4 | TM6BCE | Timer 6 capture B operation enable flag |

Enables/disables caputure operation for TM6CB

0:　　Disables caputure operation. (Pin input is ignored.)

1:　　Enables caputure operation.

| 5 | TM6BEG | Timer 6 B pin polarity selection flag |

Selects the valid edge for the input on the TM6IOB pin, and the output polarity.

0:　　Rising edge valid

　　　Positive polarity output (Reset: "L" level; set: "H" level)

1:　　Falling edge valid

　　　Negative polarity output (Reset: "H" level; set: "L" level)

| 6 | TM6BM0 | Timer 6 compare/capture B operating mode setting flag (LSB) |
| 7 | TM6BM1 | Timer 6 compare/capture B operating mode setting flag (MSB) |

These bits set the TM6CB operating mode

00:　　Compare register (single-buffer)

01:　　Compare register (double-buffer)

10:　　Capture register (single-edge operation)

11:　　Capture register (dual-edge operation)

When dual-edge capture mode is set, the setting of TM6BEG is ignored.

9.6.7 Timer 6 Compare/Capture Register A

Register symbol: TM6CA

Address: x'340010C4

Purpose: This register is the compare/capture register A for timer 6.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM6 CA15	TM6 CA14	TM6 CA13	TM6 CA12	TM6 CA11	TM6 CA10	TM6 CA9	TM6 CA8	TM6 CA7	TM6 CA6	TM6 CA5	TM6 CA4	TM6 CA3	TM6 CA2	TM6 CA1	TM6 CA0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) If set as compare register

Interrupts is generated when TM6BC equal to TM6CA.

Cycle of timer 6 can be set to (setting value +1) by clearing TM6BC when it matches with TM6CA.

(2) If set as compare register of double buffer

The data written to TM6CA will be stored into buffer for a moment.

This might cause reading of the previous value rather than the new one when the reading occur just after writing to TM6CA.

Under following condition, value in buffer is loaded into compare register.

1) Initialization of timer 6

2) When overflow occurs. (In the case of TM6CA=0)

3) When TM6BC matches with TM6CA. (in the case of TM6CAE = 1)

(3) If set as capture register

If TM6IOA pin gets edge selected in TM6AEG flag, value of TM6BC is captured into TM6CA, and then an interrupt occurs.

Under both edge capture setting, capturing happens at either of rising or falling edge, and interrupt occurs.

9.6.8 Timer 6 Compare/Capture Register B

Register symbol: TM6CB

Address: x'340010D4

Purpose: This register is the compare/capture register B for timer 6.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TM6 CB15	TM6 CB14	TM6 CB13	TM6 CB12	TM6 CB11	TM6 CB10	TM6 CB9	TM6 CB8	TM6 CB7	TM6 CB6	TM6 CB5	TM6 CB4	TM6 CB3	TM6 CB2	TM6 CB1	TM6 CB0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1)If set as compare resister

Interrupts occur whenever TM6BC matches with TM6CB.

(2)If set as compare register of double buffer

The value written to TM6CB is stored into buffer once. This may cause reading of previous value when reading happens after writing to TM6CB.

Under following condition, the value is loaded from buffer to compare register.

Then TM6BC is set to x'0000.

    1) At the initialization of timer 6.

    2) When overflow occurs. (in the case, TM6CAE = 0)

    3) TM6BC matches with TM6CA.

       (in the case, TM6CA is compare register setting and TM6CAE = 1)

    4) When TM6CA is captured. (in the case, TM6CA is capture register setting and TM6CAE = 1)

(3)If set as capture register

If TM6IOB pin gets edge selected in TM6BEG flag, value of TM6BC is captured into TM6CB, and then an interrupt occurs.

Under both edge capture setting, capturing happens at either of rising or falling edge, and interrupt occurs.

9.6.9   Prescaler Control Register

Register symbol: TMPSCNT

Address: x'34001071

Purpose: Controls prescaler operations.

Bit No.	7	6	5	4	3	2	1	0
Bit name	TMPS CNE	—	—	—	—	—	—	—
When reset	0	0	0	0	0	0	0	0
Access	R/W	R	R	R	R	R	R	R

Bit No.	Bit name	Description
0 to 6	_	Always returns "0".
7	TMPSCNE	Prescaler operation enable flag

Enable/disables 1/8 IOCLK amd 1/32 IOCLK prescaler operation.

0: Prescaler operation disable.

1: Prescaler operation enabled.

This prescaler also serves as the 1/8 IOCLK or 1/32 IOCLK that is used by 8-bit timers.

9.7 Description of Operation

9.7.1 Timers 4 and 5

Timer 4 and 5 have internal register for initialization, and down counter.

Available as both interval timer and event counter.

9.7.1.1 Interval Timer, Timer Output

Use following setting for the case that timer 4,5 is used as a interval timer.

Each timer works as interval timer to generate an interrupts at the specified cycle.

(Refer to figure 9-7-1,9-7-2,9-7-3)

Please refer cascaded connection at section 9.7.1.3 to use them as a 32-bit timer by cascading.

Steps for start up

(1) Set the division ratio

Give division ratio into TMnBR.

The interrupting cycle = (TMnBR value + 1) X clock source cycle.

(2) Select clock source

Give clock source into TMnCK[2:0] of TMnMD register.

To use 1/8 IOCLK or 1/32 IOCLK as a clock source, allow the prescaler operation by setting 1 for

TMPSCNE of TMPSCNT before the permission of the counter operation of Timer 4,5

(3) Timer initialization

Initialize timer n by setting 1 into TMnLDE of TMnMD register.

Timer output is reset by loading TMnBR value into TMnBC as initial value.

Must switch to the normal operation mode by setting 0 into TMnLDE just after the initialization.

(4) I/O port setting (case of using the timer output)

Set output for IO port by handling I/O port register.

Please refer the chapter of I/O port for register setting.

(5) Allow count operation of timer

Count operation starts by setting 1 into TMnCNE of TMnMD register.

Underflow interrupt is generated with constant cycle with counter operation permission.

Each interrupt cause the inverting of the pin output and the loading of TMnBR value into TMnBC.

Changing of the interrupt cycle can be done by loading of TMnBR register value as initial value at the first

underflow, if the TMnBR register value is replace during the count operation.

Steps for operation termination.

(1) Stop the count operation of timer

Count operation will stop when 0 is given to TMnCNE register of TMnMD register.

(2) Initialization of timer if it is necessary

Timer output reset is same as loading TMnBR value into TMnBC as initial value when 1 is set to TMnLDE of TMnMD register.

Binary counter and pin output keep the working condition unless TMnLDE is set to 1 after timer termination.

I.e. count can be restart from the same condition before the timer is stopped if TMnCNE is set to 1 again.
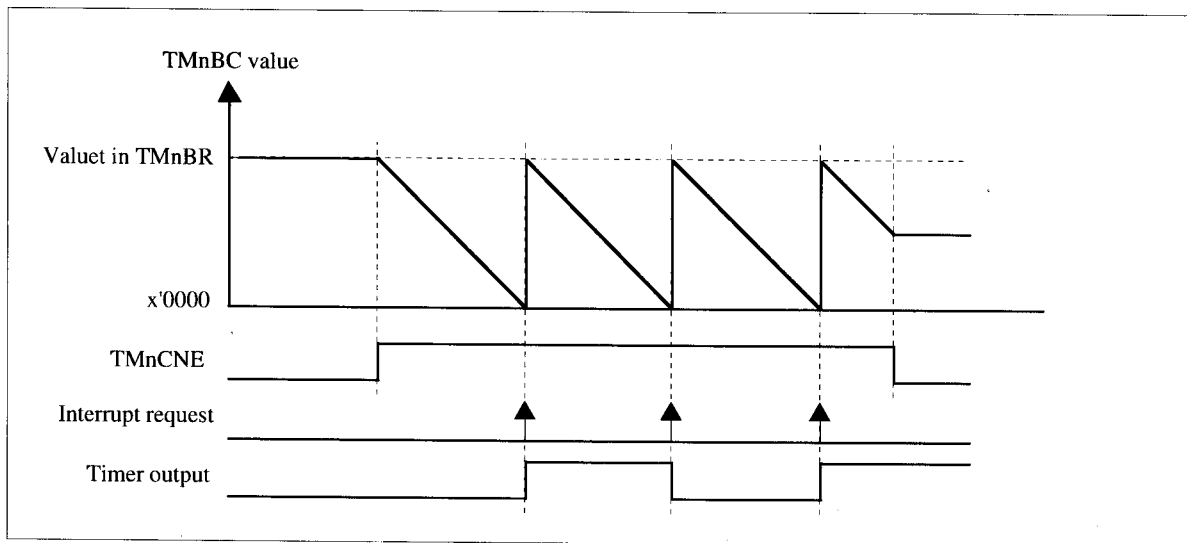
Fig. 9-7-1 Interval Timer Operation
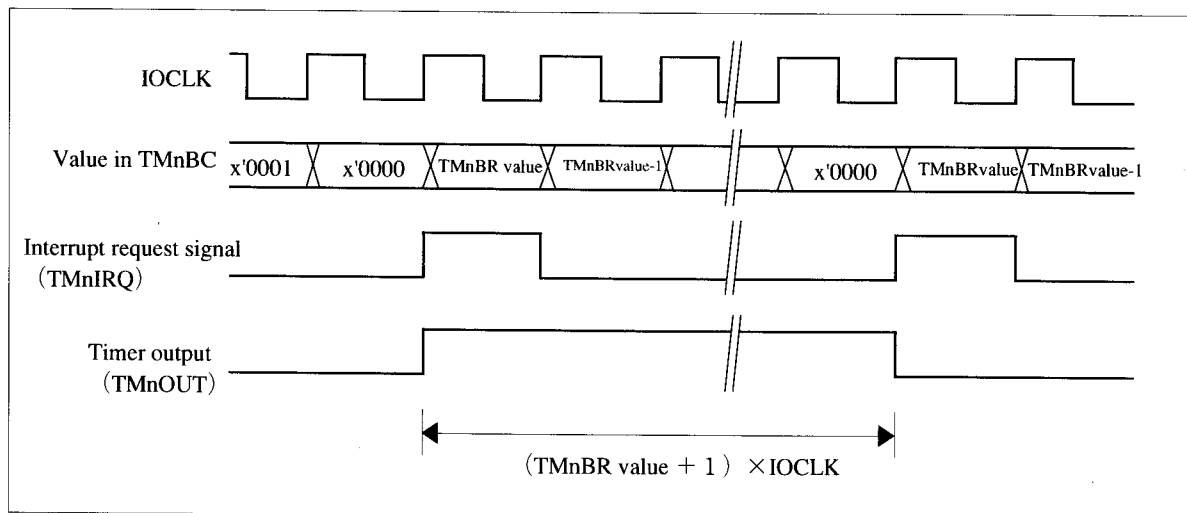
Fig. 9-7-2 Interval Timer Operation (When Clock Source = IOCLK)
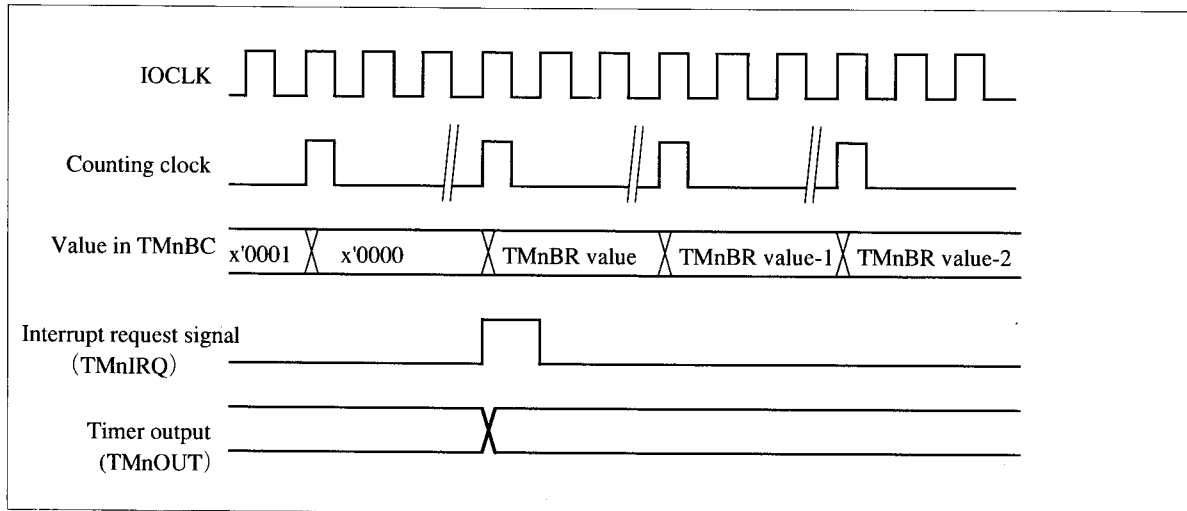
Fig. 9-7-3 Interval Timer Operation (When Use Prescaler)

9.7.1.2 Event Counter

Use the following setting when Timer 4,5 is used as a event counter.
Please refer to cascaded connection section 9.7.3.1 to use them as 32-bit counter by cascading.

Steps for start up

(1) Set the division ratio

Set the division ratio of TMnBR

Interrupt is generated whenever (TMnBR value + 1) counts of the pin input rising edge.

(2) Select clock source

TMnCK[2:0] of TMnMD register contains value for clock source to TMnIO pin input.

(3) Initialization of timer

Initialize timer n by giving 1 into TMnLDE of TMnMD register.

Load TMnBR value into TMnBC as initial value.

Must switch to normal operation mode by setting 0 into TMnLDE after the initialization.

(4) I/O port setting

Set I/O port to pin input.

Please refer chapter of I/O port for resister setting.

(5) Permission for the count operation

Give permission for the count operation of the timer by setting value 1 to TMnCNE of TMnMD register.

With the permission to count operation counter start to count the rising edge of the pin input. And load TMnBR value into TMnBC as the underflow of the binary counter occurs. (Fig. 9-7-4)

TMnBR register value, changed during the counter operation is performed, is loaded at the first underflow after the changing.

Steps for operation termination

(1) Stop count operation

Count operation stops when 0 is set to TMnCNE of TMnMD register.

(2) Initialization of timer if it is necessary

Load TMnBR value into TMnBC as initial value by giving 1 to TMnLDE of TMnMD register.

Binary counter keeps the operation condition value unless TMnLDE is set to 1 after the termination of the timer.

I.e. Count can be start from the interrupted condition when TMnCNE is given a value 1 again.

> **①** **Input pulse width must be more than IOCLK X 1.5 since IOCLK being sampled at pin input.**
> **Event count operation is inhibited when IOCLK is terminated (HALT, STOP mode).**
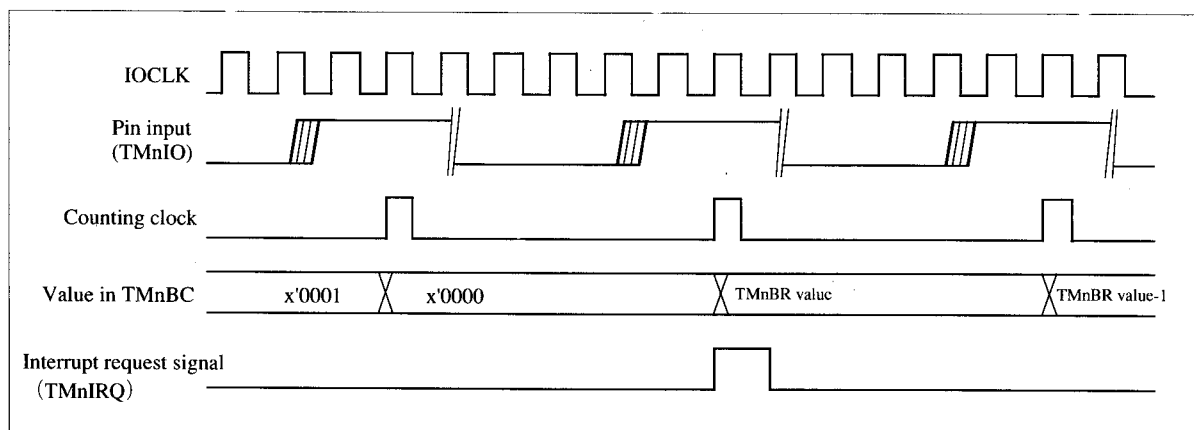
Fig. 9-7-4 Event Counting Operation

9.7.1.3 Cascaded Connection

Timer 4,5 can be used as pure 32-bit timer by cascading Timer 5 as upper half and Timer 4 as lower half. Please use the following setting to use 16-bit timer under cascaded condition.

(1) Setting of the division ratio

Set the division ratio into TMnBR.

e.g. Case timer 4, 5 is used as 32-bit timer and interrupt cycle is set as x'12345678:

To have the given interrupt cycle, TMnBR must be set as x'12345677, x'12345678-1.

The lower half TM4BR has value x'5677 and the upper half TM5BR has value x'1234.

Since 32-bit access is available, TM4BR and TM5BR can be set by one instruction simultaneously.

To change the TMnBR value during the counter operation, please change them by one instruction simul taneously.

(2) Select clock source

Set the specified clock source into the lower half of timer, Timer 4.

Set the cascaded connection for clock source into the upper half of timer, Timer 5.

(3) Initialization of the timer

Initializing can be done by set 1 to TMnLDE flag of both timer 4, 5.

(It's not necessary to done setting for timer 4,5 simultaneously.)

(4) Permission for the count operation

To permit, please use one the following step.

1) First give count operation permission to upper half, Timer 5, then give permission to the lower half, Timer 4.

2) Give permission to Timer 4 & 5 simultaneously.

(5) Termination of count operation

For operation termination, please use one of the following.

1) First terminate count operation of the lower half, Timer 4, then do same for upper half, Timer 5.

2) Terminate both Timer 4, 5 simultaneously.

(6) Timer output and Interruption

Timer output and interruption is available only for the upper half of the timer, Timer 5.

Timer output and interruption to for the lower half, Timer 4, is out of the guarantee.

Fig. 9-7-5 shows the operation of Timer 4,5 under cascaded connection condition. (Select IOCLK as clock source)

Under condition "TM5B≠ x'0000", after TM4BC become underflow, TM4BC is set x'FFFF and TM5BC is decremented by one.

Under condition "TM5BC = x'0000", after TM4BC become underflow, the value of TM4BR and TM5BR are loaded into TM4BC and TM5BC respectively, and interrupt of timer5 occurs.
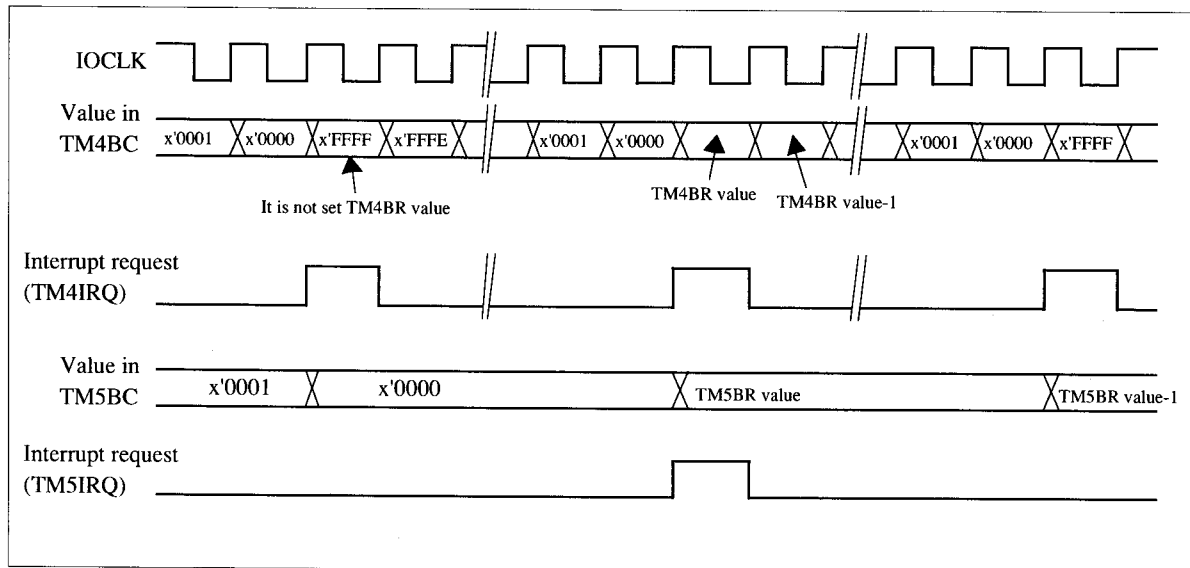
Fig. 9-7-5 Operation on Cascaded Connection

9.7.2 Timer 6

Timer 6 equips an up counter and 2 compare/capture register.

Each compare/capture registers can be used as either of a compare register or a capture register respectively.

9.7.2.1 Compare Register Setting

Please take the following steps before the initialization of the timer 6 to use compare/capture register A, B of timer 6 as a compare register.

For now compare/capture register A is used in the example, but the steps are same for compare/capture register B.

(1) Set the mode of compare/capture register A

Set TM6MDA as following:

TM6AO2,1,0	optional	
TM6ACE	0	: Inhibit capture operation.
TM6AEG	optional	
TM6AM1,0	0 0	: Compare register (single buffer) ,or
	0 1	: Compare register (double buffer)

Please select double buffer mode if the compare register's value need to be changed during the count operation.

(2) Give value, to be compared with, into compare/capture register A

Set value into TM6CA

If the setting is double buffer, the specified value won't be loaded into compare/capture register at this moment.

Even in the case of the reading of TM6CA, the previous value is read.

The specified value is loaded when timer 6 is initialized.

Fig 9-7-6 shows the timing of an interrupt for compare/capture register A under counter operation of the timer 6.

If it's set to double buffer mode the value is loaded from buffer into compare/capture register A with the timing of clearing TM6BC.
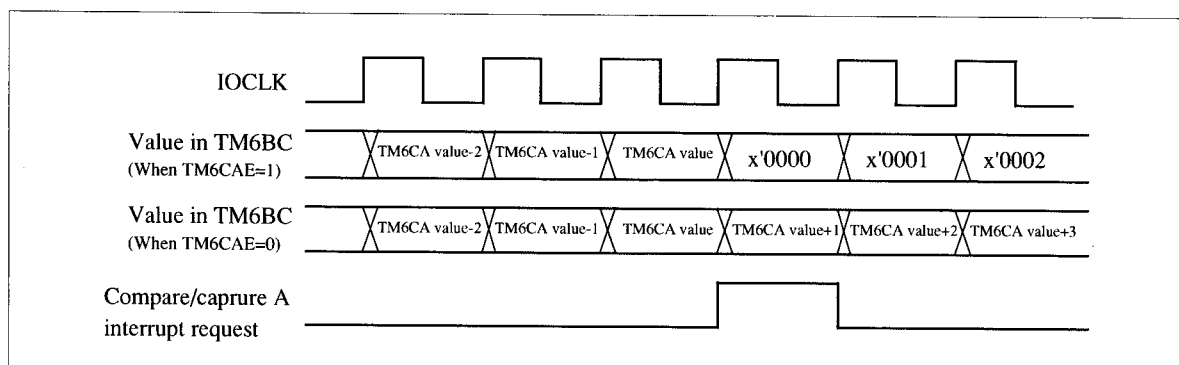
Fig. 9-7-6 Operation of Compare Register (When Clock Source = IOCLK)

9.7.2.2 Setting for Capture Register

Take the following steps, when Timer 6 compare/capture register A,B as capture register before the timer 6 is initialized.

In this section compare/capture register A is used to explain the steps; however, steps for compare/capture register B is exactly same.

(1) Mode setting for the compare/capture register A

Set the TM6MDA as show below.

TM6AO2,1,0	optional	
TM6ACE	1	: Permission for capture operation.
TM6AEG	optional (Select either of rising edge or falling edge for single edge operation)	
TM6AM1,0	1 0	: Capture register (for single edge), or
	1 1	: Capture register (for dual edge)
		If select dual edge operation, setting of TM6AEG is ignored.

Fig 9-7-7 shows steps after count operation of the timer 6 is permitted: Show when to load the value of TM6BC into TM6CA. At the next step, interrupt of compare/capture register A occurs. (Capturing won't happen unless the count operation is performing.)

If dual edge are selected as reading timing, capturing occurs whenever wither of rising edge or falling edge is produced.

It's hard to tell which edge does input has. (It is possible to read the electrical level of the pin input.)

By setting 0 to TM6ACE, capturing can be avoid even during the count operation.

If TM6CAE of TM6MD register is set to 1 and if TM6CA is set as a capture register, TM6BC is cleared with the capturing operation of TM6CA.

At this moment, value in the buffer is load into compare register if TM6CB has double buffer compare register setting. The output pattern of the additional bits is as shown below.
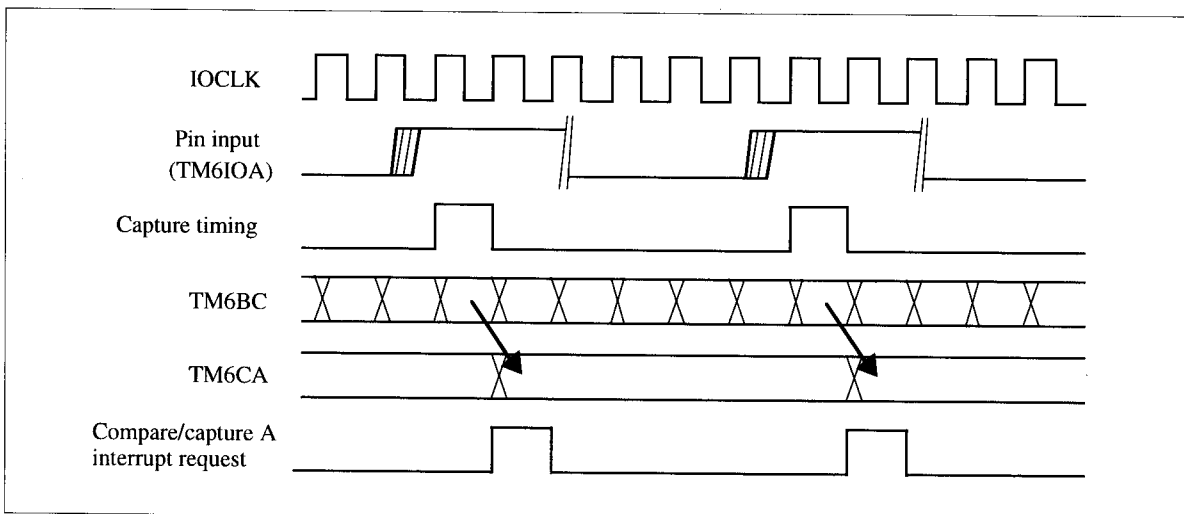
Fig. 9-7-7 Operation of Input Capture (When Select Rising Edge)

9.7.2.3 Setting of Pin Output

Any type of output wave is available for TM6IOA, TM6IOB pin using timer 6.

(1) Setting for the initial output level

Initializing timer 6 with the setting, TM6LDE flag of TM6MD = 1, cause TM6IOA pin output level to be the value in TM6AEG flag of TM6MDA register.

Similarly TM6IOB pin output level is value in TM6BEG flag of TM6MD register.

Once switched to the normal operating mode, set 0 for TM6LDE flag of TM6MD register, the pin output level never changes even the TM6AEG/TM6BEG flag is changed.

(2) Setting for output wave during count operation.

TM6IOA pin has a wave, set into TM6AO0,1,2 flag, as output wave when count operation of timer 6 is permitted by setting TM6CNE flag of TM6MD to 1.

Similarly TM6IOB pin output wave equal to the wave set in TM6BO0,1,2 flag of TM6MDB register.

TM6AEG/ TM6BEG flag setting is used only to change the output level.

I.e. Output level stays until the next output is changes after the setting is replaced.

Example of TM6IOA pin output wave is shown below. They are exactly same for TM6BIO.

Fig. 9-7-8 shows the TM6IOA pin output wave in the case "Simultaneously set and reset for TM6BC and TM6CA." Reset has higher priority than set if the set and reset are generated simultaneously.
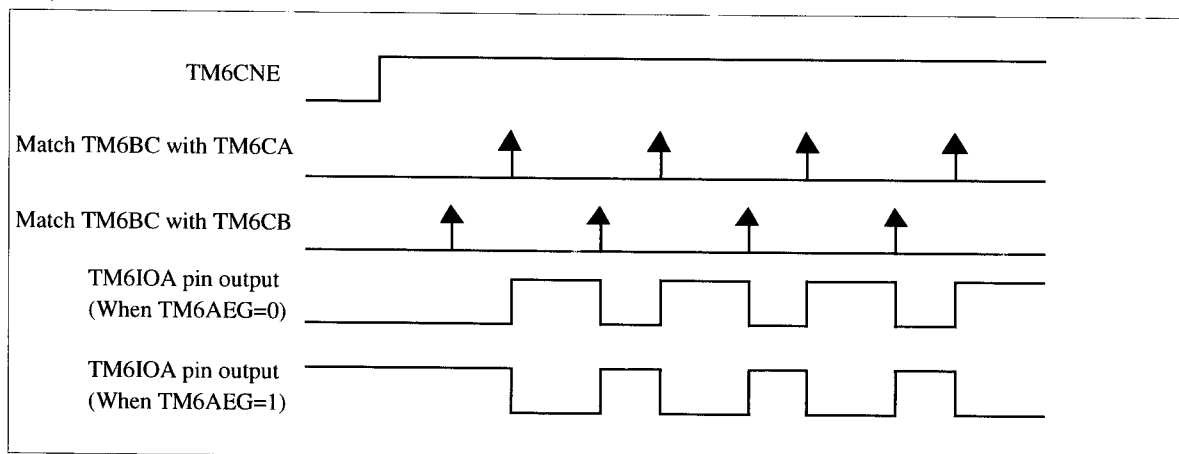
Fig. 9-7-8 Pin Output Waveform (1)

Fig 9-7-9 shows TM6IOA pin output wave in the case "Simultaneously set for TM6BC and TM6CA, and reset when overflow of TM6BC occurs." Reset has higher priority than set when both reset and set are generated simultaneously.
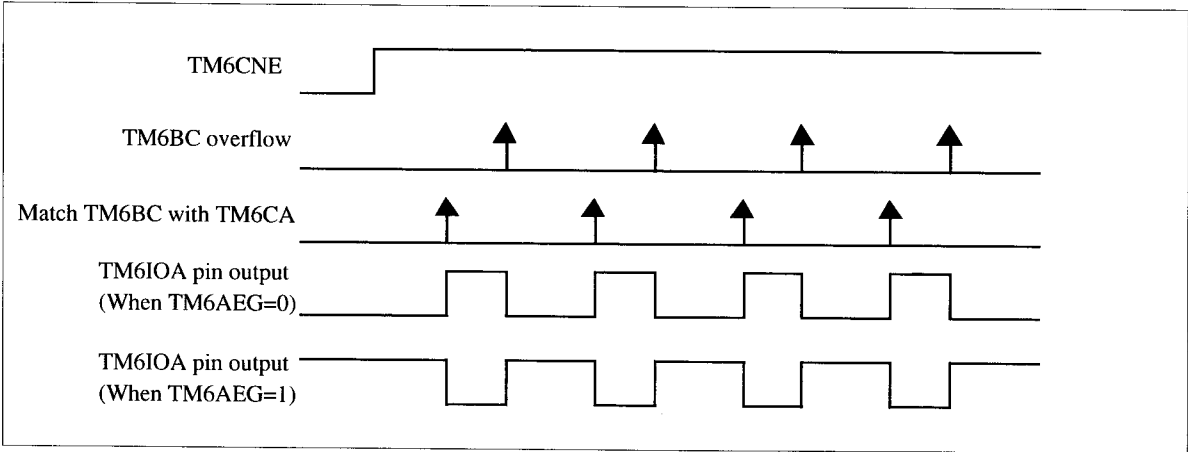
Fig. 9-7-9 Pin Output Waveform (2)

Fig 9-7-10 shows TM6IOA pin output wave in the case "Simultaneously set for TM6BC and TM6CA."
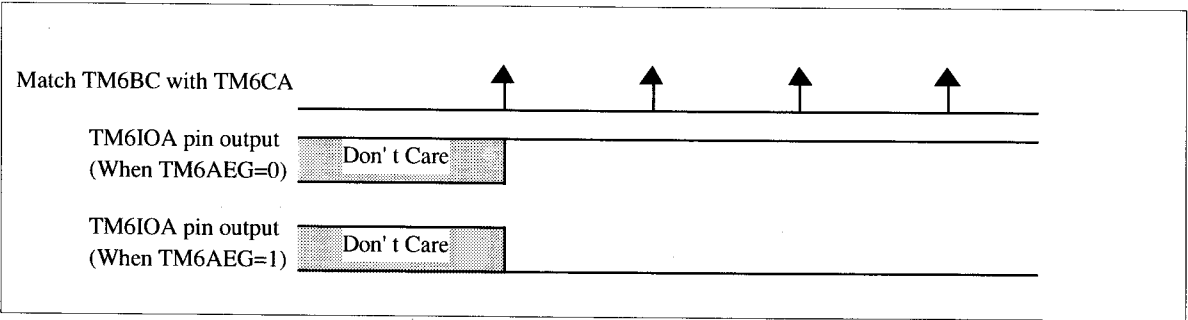
Fig. 9-7-10 Pin Output Waveform (3)

Fig 9-7-11 shows TM6IOA pin output wave in the case "Simultaneously reset for TM6BC and TM6CA."
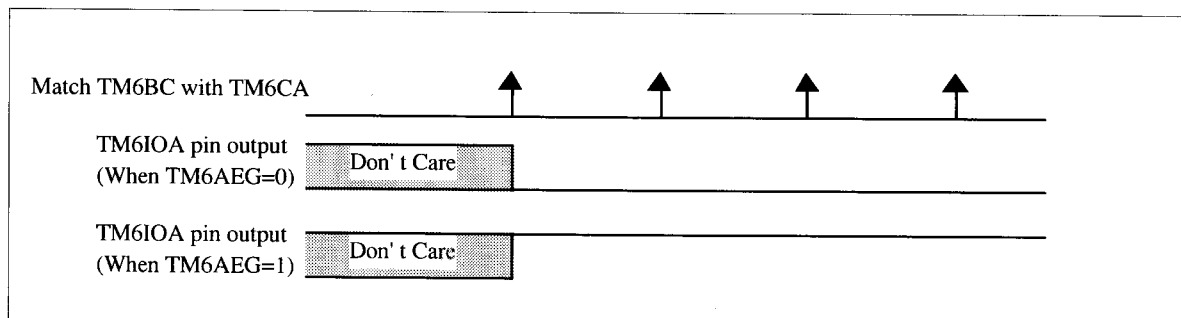
Fig. 9-7-11 Pin Output Waveform (4)

Fig 9-7-12 shows TM6IOA pin output wave in the case "Toggled output".
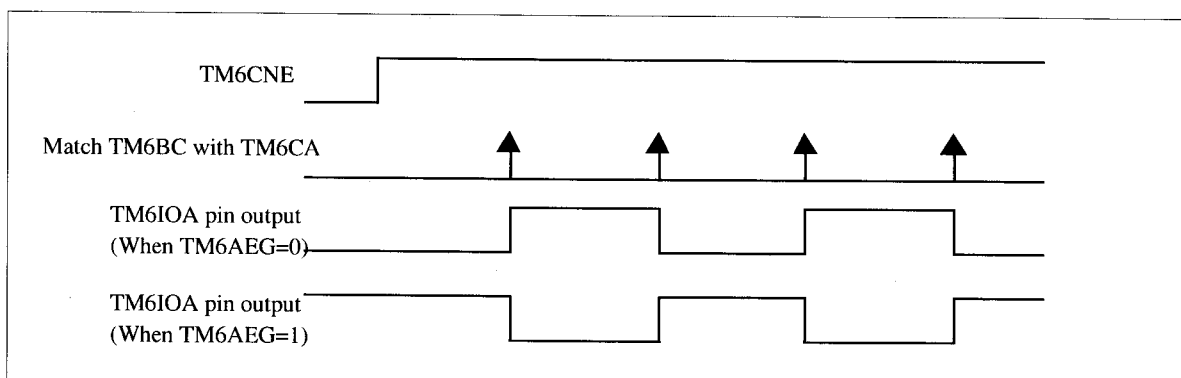
Fig. 9-7-12 Pin Output Waveform (5)

9.7.2.4 Start Up by an External Trigger

Timer 6 can be start up with the input to TM6IOB pin.

Fig. 9-7-13 shows the signal wave at the start up.

Compare/capture register A,B can be use as either of compare register or capture register.

Steps for start up

(1) Select the input edge

Select input edge by setting TM6BEG of TM6MDB.

Start up edge is oppose to the normal setting:

When TM6BEG = 0, Falling edge of input causes starting up.

When TM6BEG = 1, Rising edge of input causes starting up.

(2) Setting for the operation mode

Setting for TM6MD register is shown as follows

TM6CK2,1,0	optional	: Select any clock source.
TM6CAE	optional	: "1" when setting one-shot operation and the interrupt cycle.
TM6ONE	optional	: "1" for one-shot operation.
TM6TGE	0	: This disables timer start by an external trigger.
TM6PM1,0	optional	: This setting is ignored.
TM6PME	0	: Selects the normal waveform.
TM6LDE	0	: Normal operation.
TM6CNE	0	: Stops counting operation.

Permission for prescaler operation, set 1 into TMPSCNE of TMPSCNT register, must be done before the permission for count operation of timer 6 when 1/8 IOCLK or 1/32 IOCLK is used as clock source.

(3) Initialization of timer

Initialize timer 6 by set 1 into TM6LDE of TM6MD register.

Clear the TM6BC. Reset pin output. When compare/capture register has double buffer compare setting, load the value from buffer into compare register.

Switch to the normal mode, set 0 into TM6LDE, must be done after the initialization.

(4) Setting for I/O port

Set TM6IOB as input pin. Setting of TM6IOA pin is optional.

Please refer the chapter of I/O port for the register setting.

(5) Permission for starting up of timer with external trigger

Set 1 into TM6TGE of TM6MD register.

Timer 6 starts up when specified edge is give to TM6IOB. (TM6CNE flag of TM6MD register is set by hardware.)

Steps for operation termination

    (1) Inhibit starting up of timer by external trigger

        Set 0 into TM6TGE of TM6MD register.

    (2) Stop count operation

        Set 0 into TM6CNE of TM6MD register.

Setting 0 to TM6TGE and TM6CNE simultaneously might cause resetting of TM6CNE due to the pin input timing. To avoid this case, user must set 0 into TM6TGE first then set 0 to TM6CNE.
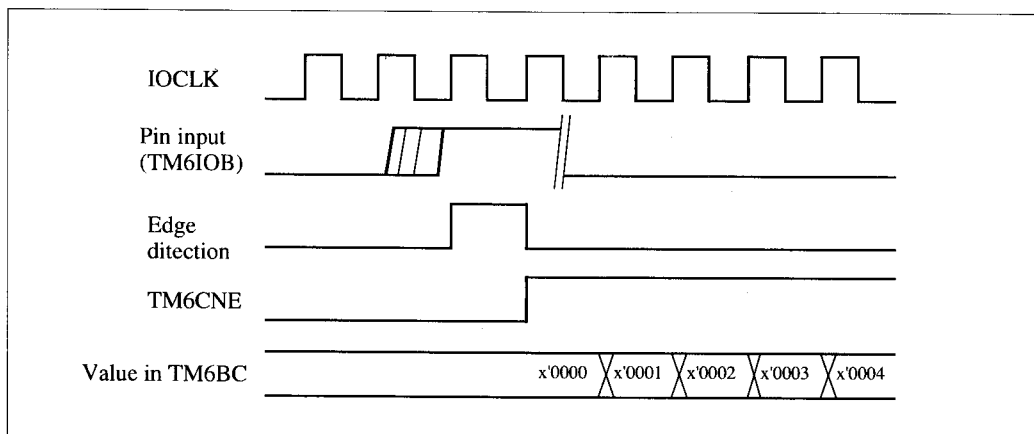
Fig. 9-7-13 Start Up by an External Trigger (When rising edge is selected)

9.7.2.5 One Shot Operation

Timer 6 can be terminating by the matching of TM6BC and TM6CA.

Fig. 9-7-14, 15 shows the signal wave while timer 6 is inactivating.

Compare/capture register B can be used as either of compare register or capture register.

Steps for start up

(1) Setting of compare/capture register A

Set TM6MDA register as shown below.

TM6AO2,1,0	optional	
TM6ACE	0	: Inhibit capturing operation
TMAEG	optional	
TM6AM1,0	0 0	: Compare register (single buffer) or
	0 1	: Compare register (double buffer)
		Must use double buffer setting if the value in compare/capture register A is changed during count operation.

(2) Set the value to terminate timer

Give the comparison value into TM6CA.

Timer is terminated after (value in TM6CA register + 1) counts.

(3) Setting of operation mode

Set TM6MD register as follows

TM6CK2,1,0	optional	: select clock source
TM6CAE	1	: Clear TM6BC if TM6CA equal to TM6BC
TM6ONE	1	: Permission for one shot operation
TM6TGE	0	: Inhibit starting up of timer by external trigger.
TM6PM1,0	optional (This setting never to be used)	
TM6PME	0	: Select normal wave
TM6LDE	0	: Normal operation
TM6CNE	0	: Terminate count operation

Prescaler operation must be permitted, setting 1 to TMPSCNE of TMPSCNT register before the count operation of timer 6 is allowed when 1/8 IOCLK or 1/32 IOCLK is used as clock source.

(4) Initialization of timer

Initialize timer 6 by set 1 into TM6LDE of TM6MD register.

Clear the TM6BC. Reset pin output. When compare/capture register has double buffer compare setting, load the value from buffer into compare register.

Switch to the normal mode, set 0 into TM6LDE, must be done after the initialization.

(5) Set the I/O port (when using pin output)

Set the I/O port to "timer output pin".

In the I/O port register, select "timer output" for the output signal and then set the output pin.

Refer to chapter of I/O port about register setting.

When the timer is to be started up by an external trigger, set the TM6IOB pin to "input pin".

(6) Enable the timer counting operation

The counting operation starts when the TM6CNE in the TM6MD register is set to "1".

If the timer is to be started by an external trigger, leave TM6CNE set to "0" and set TM6TGE to "1".

When starting to count up again after TM6BC and TM6CA have matched, the hardware clears the TM6CNE flag to stop the counting operation. TM6BC is also cleared.

Steps for operation termination. (When use output pin)

(1) Inhibit "Starting up of timer by external trigger."

Set 0 into TM6TGE of TM6MD register.

(2) Terminate count operation

Set 0 into TM6CNE of TM6TGE register.

Setting 0 to TM6TGE and TM6CNE simultaneously might cause resetting of TM6CNE due to the pin input timing. To avoid this case, user must set 0 into TM6TGE first then set 0 to TM6CNE.
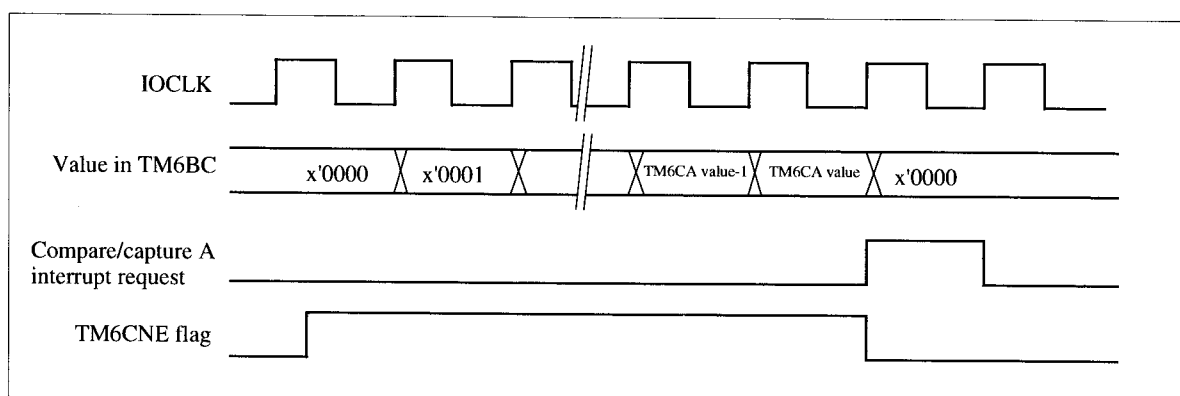
Fig. 9-7-14 One-shot Operation (When Clock Source = IOCLK)
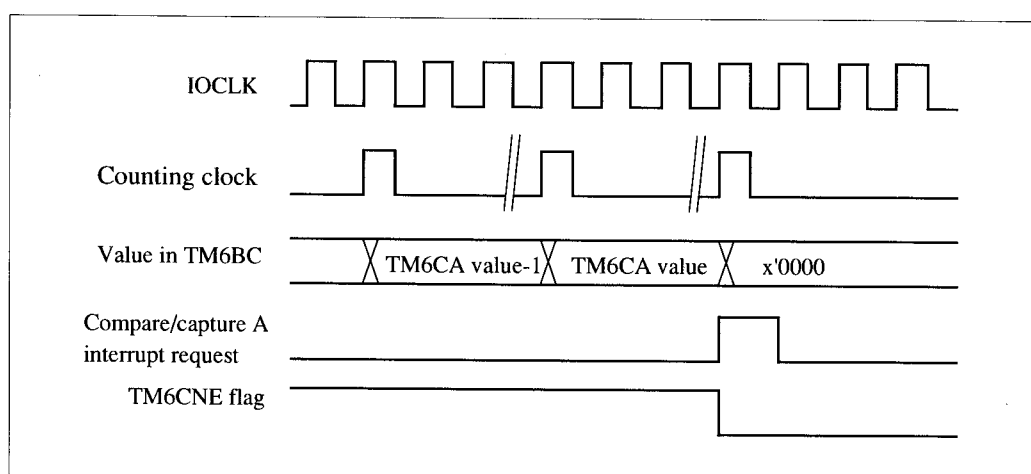
Fig. 9-7-15 One-shot Operation (When Using Prescaler)

9.7.2.6 Interval Timer

Please take the following steps to use timer 6 as an interval timer.

Timer 6 works as interval timer, generate interrupt of compare/capture register A with specified cycle. (fig. 9-7-16,17)

Compare/capture register B is available as either of compare register or capture register.

Please refer section 9.7.2.1 for compare register setting, and section 9.7.2.2 for capture register setting.

Steps for start up

(1) Setting mode of compare/capture register A

Setting for TM6MDA register.

TM6AO2,1,0	optional	
TM6ACE	0	: Inhibit capture operation
TMAEG	optional	
TM6AM1,0	0 0	: Compare register (single buffer) or
	0 1	: Compare register (double buffer)
		Must use double buffer if interrupt cycle is changed during the count operation.

(2) Set the division ratio

Give division ratio into TM6CA

Interrupt cycle of compare/capture register A= (value in TM6CA + 1) X clock source cycle.

(3) Setting for operation mode

Setting of TM6MD register.

TM6CK2,1,0	optional	: Select clock source
TM6CAE	1	: Clear TM6BC if TM6BC matches with TM6CA.
TM6ONE	0	: Inhibit one shot operation.
TM6TGE	0	: Inhibit starting up of timer by external trigger.
TM6PM1,0	optional (No effects to operation.)	
TM6PME	0	: Select normal wave
TM6LDE	0	: Normal operation
TM6CNE	0	: Terminate count operation
		Permit prescaler operation, setting 1 into TMPSCNE of TMPSCNT register, before the permission for count operation of timer 6 when 1/8IOCLK or 1/32 IOCLK is used as a clock source.

(4) Initialization of timer

Initialization of timer 6, setting 1 into TM6LDE of TM6MD register.

TM6BC is cleared, and pin output is reset.

If TM6CA has double buffer compare register setting, value in buffer is load into compare register.

User must switch to normal operation mode, setting 0 is not TM6LDE, after the initialization.

(5) Setting for I/O port (case using pin output)

Set I/O port to timer pin output.

Select timer pin output as output signal with I/O port register.

Please refer chapter of I/O port for register setting.

(6) Permission for count operation

Count operation starts by setting 1 into TM6CNE of TM6MD register

Interrupt request of compare/capture A is generated with a specified cycle when count operation is permitted.

Changing of the interrupt cycle is available by handling the value in the TM6CA register, causing the loading from buffer to compare register when TM6BC is cleared.

TM6CA must be set to double buffer compare register to change the interrupt cycle during count operation.

Steps for operation termination

(1) Terminate count operation

Count operation is stopped, if set value 0 into TM6CNE of TM6MD register.

(2) Initialize timer as necessary

Setting 1 into TM6LDE of TM6MD register cause the clearing of TM6BC and the reset of timer output.

If TM6CA register has double buffer setting, value in compare register buffer is loaded to compare register.

Binary counter, compare register and pin output keep the level before the termination, unless TM6LDE is set to 1 after the termination.

I.e. continuing of the count from the terminated condition is available by setting 1 into TM6CNE.
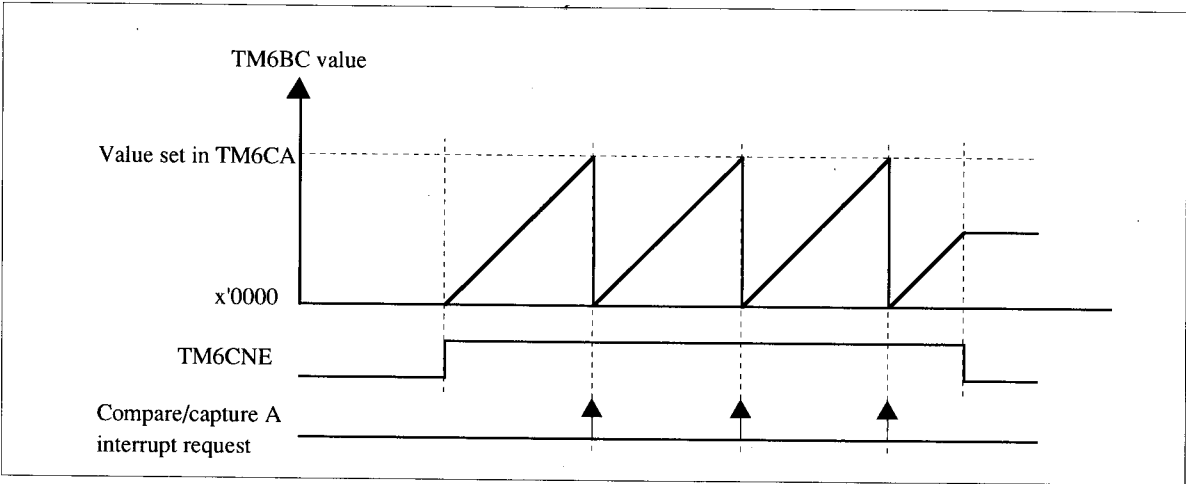
Fig. 9-7-16 Operation of Interval Timer on Timer 6 (1)
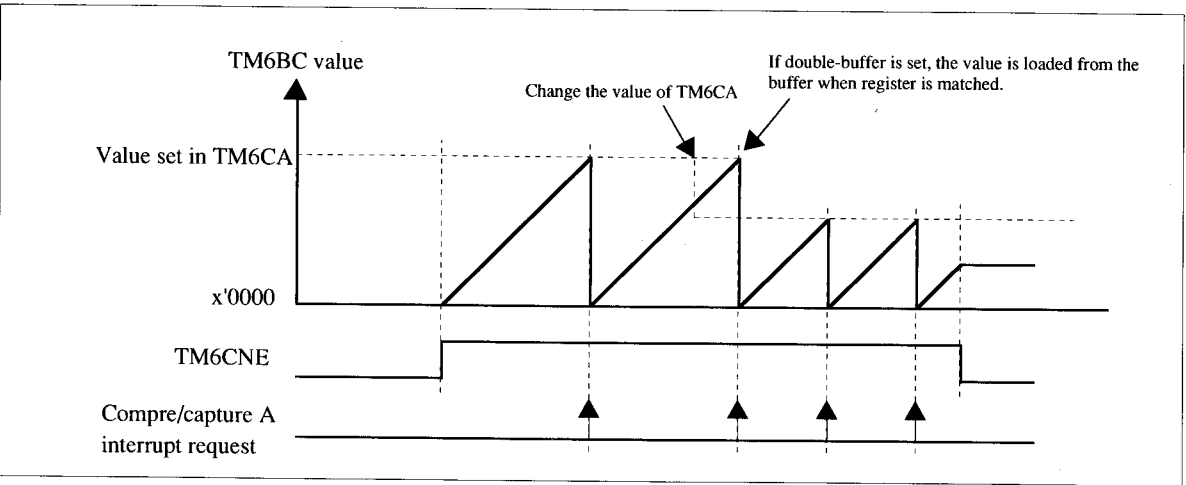
Fig. 9-7-17 Operation of Interval Timer on Timer 6 (2)
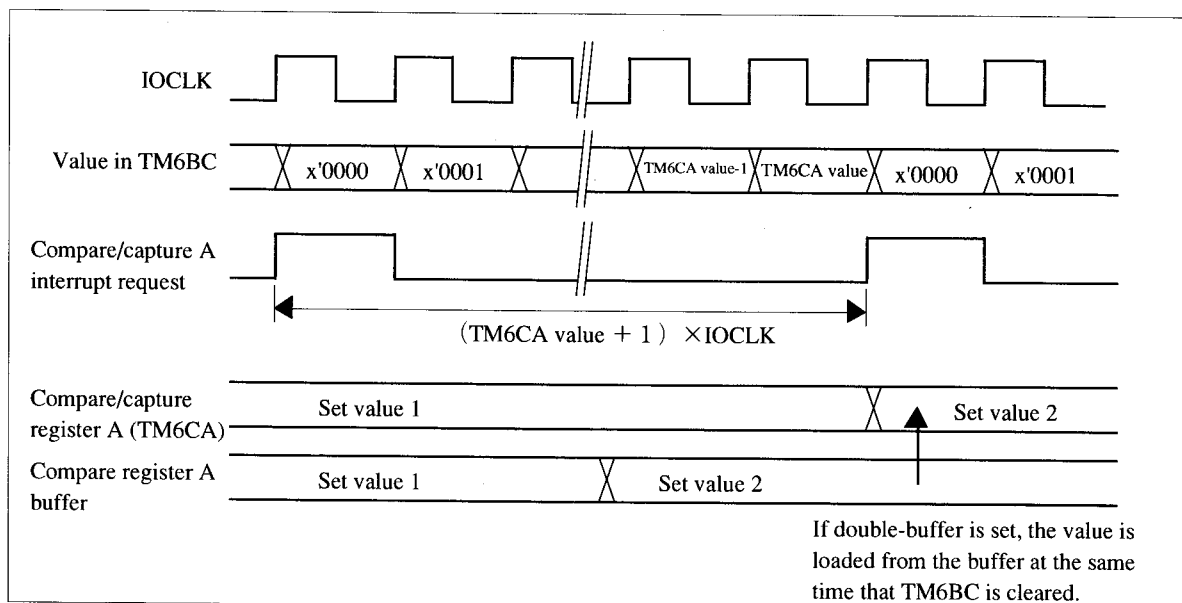
Fig. 9-7-18 Operation of Interval Timer on Timer 6 (When Clock Source = IOCLK)
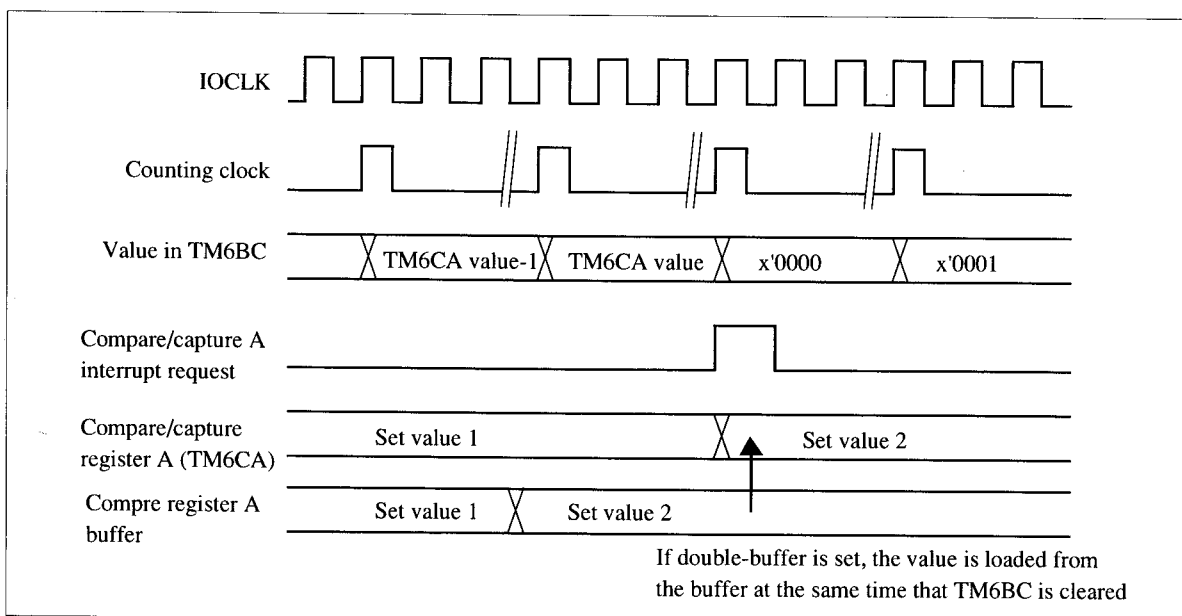
Fig. 9-7-19 Operation of Interval Timer on Timer 6 (When Using Prescaler)

9.7.2.7 Event Counter

Please take the following steps to use timer 6 as event counter.

Interrupt of compare/capture register A is generated whenever the specified number of edges are counted. (fig. 9-7-20)

Compare/capture B is available as compare register. Please refer section 9.7.2.1, setting of compare register.

Steps for start up
- (1) Setting of compare/capture register

  Setting of TM6MDA register

TM6AO2,1,0	optional	
TM6ACE	0	: Inhibit capture operation
TM6AEG	optional	
TM6AM1,0	0 0	: Compare register (single buffer) or
	0 1	: Compare register (double buffer)

- (2) Select input edge of TM6IOB pin

  Select either of rising/falling edge by TM6AEG of TM6MDB register.

- (3) Set division ratio

  Set division ratio into TM6CA

  Interrupt of compare/capture register A is generated when selected edge is counted (value in TM6CA + 1) times at TM6IOB.

- (4) Setting for operation mode

  Set the TM6MD register as described below

TM6CK2,1,0	111	:Sets the TM6IOB pin input as the clock source.
TM6CAE	1	:Clears TM6BC when TM6CA matches TM6BC.
TM6ONE	0	:Disables one-shot operation.
TM6TGE	0	:Disables timer start by an external trigger.
TM6PM1,0	optional	(This setting is ignored.)
TM6PME	0	:Selects the normal waveform.
TM6LDE	0	:Normal operation.
TM6CNE	0	:Stops counting operation.

- (5) Initialization of timer

  Initialize timer 6 by setting 1 into TM6LDE of TM6MD register.

  TM6BC is cleared and pin output is reset. If TM6CA has double buffer compare register setting, value in the buffer is loaded into compare register.

  User must switch to normal mode, setting 0 into TM6LDE, after the initialization.

- (6) Setting for I/O port

  Set I/O port to input pin.

  Please refer the chapter of I/O port for the register setting.

- (7) Permission for count operation

  Setting 1 into TM6CNE of TM6MD register causes the starting up of count operation.

By permitting count operation, interrupt request of compare/capture register A is generated whenever TM6IOB counts the specified edge (value in compare/capture register A + 1) times.

TM6BC is up counted with the specified edges and is cleared at the (value in compare/capture register A + 1) times counts.

Changing of the interrupt cycle is available by changing TM6CA register value during count operation, causing to load value in buffer into compare/capture register at the fist time TM6BC is cleared.

TM6CA must have double buffer setting to allow the interrupt cycle changing.

Steps for operation termination

(1) Terminate the count operation of the timer

Set 0 into TM6CNE of TM6MD register.

(2) Initialize timer if it's necessary

Setting 1 into TM6LDE of TM6MD register causes the clearing of TM6BC and the reset of the timer output.

If TM6CA register has double buffer setting, value in compare register buffer is loaded into compare register.

Binary counter, compare register and pin output keeps the condition before the termination unless TM6LDE is set to 1 after the termination of the timer.

I.e. it is possible to start recount from the terminated condition by setting 1 into TM6CNE again.

> **!** Input pulse width must be more than IOCLK X 1.5 since IOCLK being sampled at pin input.
>
> Event count operation is inhibited when IOCLK is terminated (HALT, STOP mode)
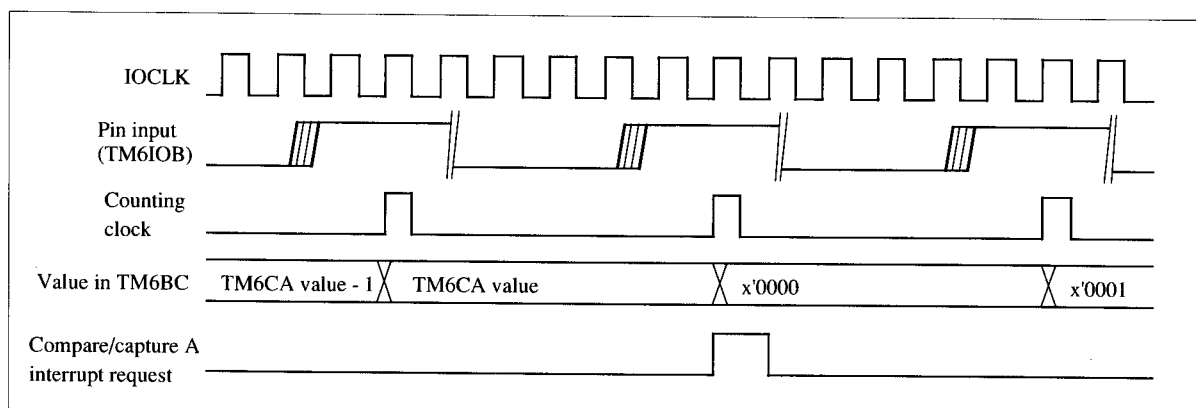
Fig. 9-7-20 Event Counting Operation (When Select the Rising Edge)

9.7.2.8   Register Settings (Additional Bit Style PWM Mode)

1. Setting TM6MD register

    While count operation is stopped, set the clock source, etc. (TM6MD register)
    Set additional bit style PWM mode and set the resolution.
    Set TM6CAE to "0".

2. Set output port

    Set the port output mode register to "timer output," set the port output control register to "output," and set the
    TM6IOA and B pins to "output."

3. Set the compare/capture register modes. (TM6MDA and B registers)

    Set compare/capture registers A and B to "compare registers (double-buffer)."
    Set the output waveform to "set when there is a match with TM6CA (TM6CB) and reset when there is an
    overflow in TM6BC."

4. Set the comparison values in TM6CA and TM6CB.

    Write the values after performing a left shift on the data so that the register MSB and the data MSB overlap.
    (In the case of 10-bit data, shift the data 6 bits to the left and write the data to bits bp15 to 6 in the TM6CA and
    TM6CB registers.)

5. Permit count operation

    Once the counting operation is enabled, the PWM waveform is output.

The basic output is an 8-bit PWM waveform, and when the clock source is set to IOCLK, the frequency is constant
at 58.4 kHz, regardless of the resolution. (When IOCLK = 15 MHz) The basic output duty ratio is determined by
the high-order 8 bits of the compare register.
The output cycle is determined by the resolution. The cycles are the same as for the corresponding 10-, 11-, 12- and
14-bit free-run counters, respectively.

Table 9-7-1 Frequencies

Resolution	Frequency for one cycle
10 bits	24.4 kHz
11 bits	12.2 kHz
12 bits	6.1 kHz
14 bits	1.5 kHz

For details on the register settings, refer to Chapter 12, "I/O Ports."

The output pattern of the additional bits is as shown below.

Table 9-7-2 Output Pattern

Low-order 8 bits of the value set in the compare register	Value of high-order 8 bits in the binary counter when additional bits are output in front of basic output
b'0XXXXXXX	b'XXXXXXX1 (1、3、5・・61、63)
b'X0XXXXXX	b'XXXXXX10 (2、6、10・・58、62)
b'XX0XXXXX	b'XXXXX100 (4、12、20・・52、60)
b'XXX0XXXX	b'XXXX1000 (8、24、40、56)
b'XXXX0XXX	b'XXX10000 (16、48)
b'XXXXX0XX	b'XX100000 (32)

When the resolution is 12 bits, the relationship between the value set in the compare register and the output waveform duty ratio is as follows.

Compare register setting	Output waveform duty ratio
X'FFFX	0
X'FFEX	$1/2^{12}$
•	
•	
X'800X	$(2^{11}-1)/2^{12}$
X'7FFX	1/2
•	
•	
X'001X	$(2^{12}-2)/2^{12}$
X'000X	$(2^{12}-1)/2^{12}$

Additional bit style PWM output
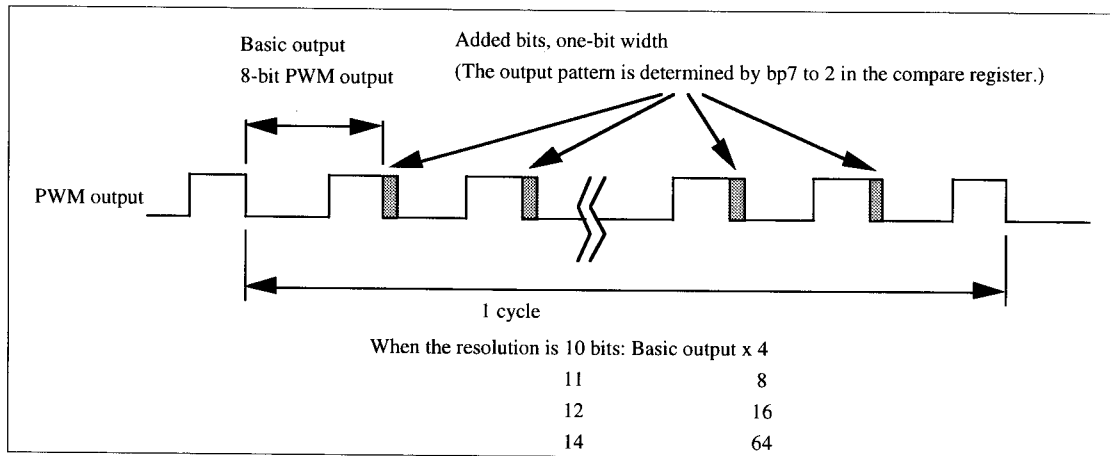
Fig. 9-7-21 Additional Bit Style PWM Output (1)

Additional bit style PWM output (when resolution is 10 bits)
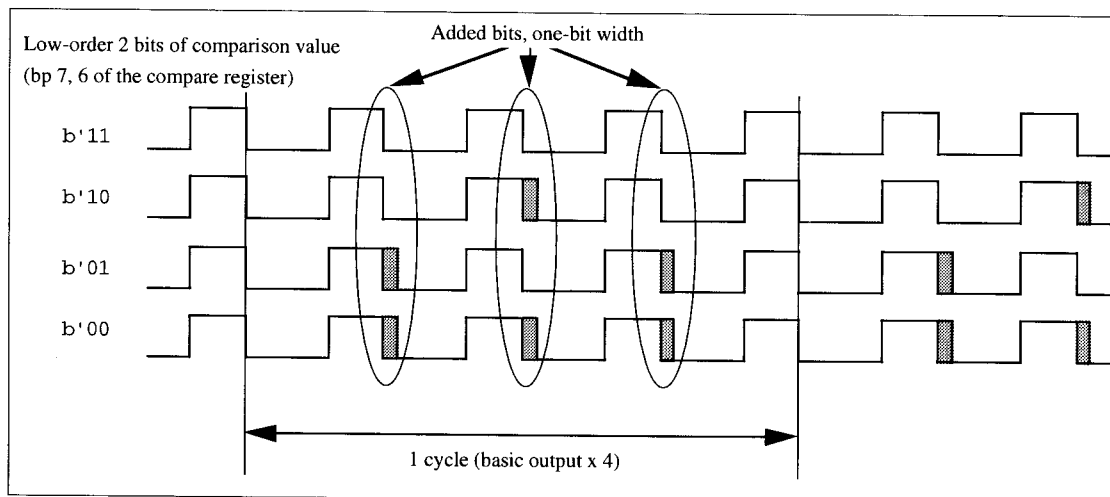
Fig. 9-7-22 Additional Bit Style PWM Output (2)

Chapter 10. Watchdog Timer

10

10.1 Overview

This microcontroller has a 25-bit binary counter built in that can be used as a 16- to 25-bit watchdog timer.

A nonmaskable interrupt and the watchdog timer overflow output can be generated in response to a watchdog timer overflow.

The watchdog timer is also used as an oscillation stabilization wait timer.

10.2 Features

● The number of bits in the binary counter is selectable.
   When the FRQS pin input is "High" (oscillating frequency: 13.0 MHz to 16.6 MHz):
      16, 18, 20, 22, or 24 bits
   When the FRQS pin input is "Low" (oscillating frequency: 26.0 MHz to 33.3 MHz):
      17, 19, 21, 23, or 25 bits
   Overflow cycle:
      3.972 ms to 1016.801 ms (when the FRQS pin input is High and the oscillating frequency is 16.5 MHz)
      3,972 ms to 1016.801 ms (when the FRQS pin input is Low and the oscillating frequency is 33 MHz)

● A nonmaskable interrupt is generated when a watchdog timer overflow occurs.

● Watchdog timer overflow output
   When an overflow occurs, the watchdog timer overflow output is output to the WDOVF pin. The watch-
   dog timer overflow output can be selected as either pulse output or level output.

● Oscillation stabilization wait time
   (when the FRQS pin input is high and the oscillating frequency is 16.5 MHz)
      When reset is released: 15.888 ms
      When recovering from STOP mode: 15.888 ms

● The chip can self-reset internally by writing the RSTCTR register.

10.3 Block Diagram
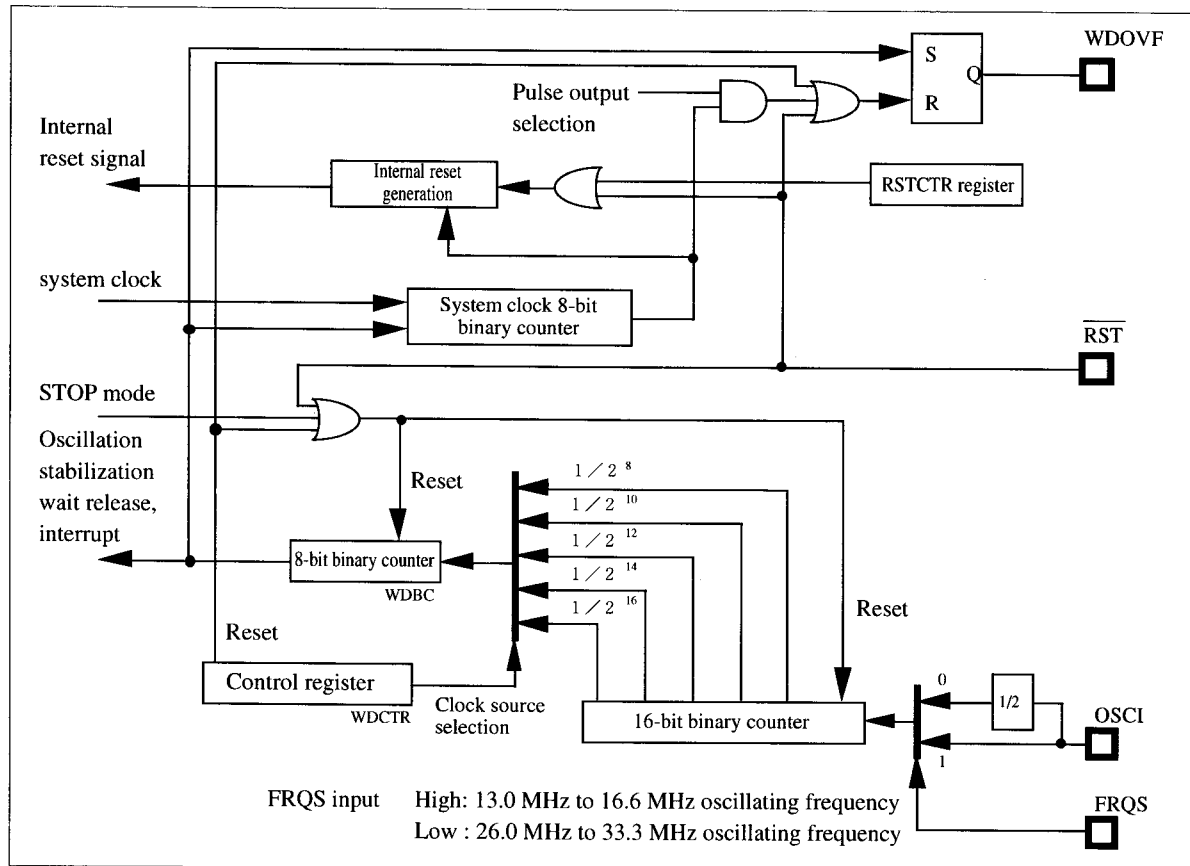
Fig. 10-3-1 Block Diagram

10.4 Description of Registers

Table 10-4-1 List of Watchdog Timer Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34004000	Watchdog binary counter	WDBC	8	x'00	8, 16
x'34004002	Watchdog timer control register	WDCTR	8	x'01	8, 16
x'34004004	Reset control register	RSTCTR	8	x'00	8, 16

10.4.1 Watchdog Binary Counter

Register symbol: WDBC

Address: x' 34004000

Purpose: Reading this counter returns the counter value of the high-order eight bits of the
                      watchdog timer.

Bit No.	7	6	5	4	3	2	1	0
Bit name	WD BC7	WD BC6	WD BC5	WD BC4	WD BC3	WD BC2	WD BC1	WD BC0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0 to 7	WDBC0 to TDBC7	Counter value of the high-order eight bits of the watchdog timer

The value that is read is not guaranteed if the value of the high-order eight bits of the watchdog timer changes while it is being read.

10.4.2 Watchdog Timer Control Register

Register symbol: WDCTR

Address: x' 34004002

Purpose: This counter sets the watchdog timer operation control conditions.

Bit No.	7	6	5	4	3	2	1	0
Bit name	WD CNE	WD RST	WD OVT	WD OVF	-	WD CK2	WD CK1	WD CK0
When reset	0	0	0	0	0	0	0	1
Access	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit No.	Bit name	Description
0	WDCK0	Clock source selection (LSB)
1	WDCK1	Clock source selection
2	WDCK2	Clock source selection (MSB)

These bits select the clock source for the high-order 8 bits of the counter.

When FRQS is high

- 000: $1/2^8$ of the OSCI input
- 001: $1/2^{10}$ of the OSCI input
- 010: $1/2^{12}$ of the OSCI input
- 011: $1/2^{14}$ of the OSCI input
- 100: $1/2^{16}$ of the OSCI input
- 101: Setting prohibited
- 110: Setting prohibited
- 111: Setting prohibited

When FRQS is low

- 000: $1/2^9$ of the OSCI input
- 001: $1/2^{11}$ of the OSCI input
- 010: $1/2^{13}$ of the OSCI input
- 011: $1/2^{15}$ of the OSCI input
- 100: $1/2^{17}$ of the OSCI input
- 101: Setting prohibited
- 110: Setting prohibited
- 111: Setting prohibited

The specific overflow cycle for each FRQS value is shown in Table 10-4-2.

<Continued>

<Continued>

Table 10-4-2 Examples of Overflow Cycle

Selection	Overflow cycle	
	When FRQS is high and oscillating frequency is 16.5 MHz	When FRQS is low and oscillating frequency is 33 MHz
0 0 0	3.972 ms	3.972 ms
0 0 1	15.888 ms	15.888 ms
0 1 0	63.550 ms	63.550 ms
0 1 1	254.200 ms	254.200 ms
1 0 0	1016.801 ms	1016.801 ms

Bit No.	Bit name	Description
3	—	Always returns "0".
4	WDOVF	Reading this bit returns the value of the WDOVF pin.
5	WDOVT	Watchdog timer overflow output selection 0: Pulse output 1: Level output
6	WDRST	Binary counter reset, watchdog timer overflow output reset 0: No reset 1: Reset
7	WDCNE	Watchdog timer count operation control 0: Count operation stopped (oscillation stabilization wait operation is possible) 1: Count operation enabled

1. **When resetting the external overflow output by writing the WDRST flag, do not simultaneously overwrite the WDOVT flag. If this flag is overwritten, the external overflow pin signal reset is not guaranteed.**

2. **When changing the value in WDCK0 to 2, first stop the watchdog timer and then reset the counter.**

10.4.3 Reset Control Register

Register symbol: RSTCTR

Address: x' 34004004

Purpose: This register is used to generate an internal reset through software.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	CHIP RST
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Bit No.	Bit name	Description
0	CHIPRST	This flag is used to generate a self-reset (internal reset). A self-reset is generated when this flag is overwritten from "0" to "1".

A self-reset is not generated if this flag is already set to "1" when it is written with a "1".

The value stored in this flag is retained even after the reset.

The CHIPRST flag is cleared either by an external reset signal or by writing a "0" to this flag through the software.

10.5 Description of Operation

10.5.1 Oscillation Stabilization Wait Operation

The watchdog timer operates as an oscillation stabilization wait timer after the reset state is released or when the microcontroller recovers from STOP mode.

The watchdog timer operates in this capacity even if the WDCNE flag is "0".

If the WDCNE flag is "1", a non-maskable interrupt is not generated even when recovering from STOP mode.
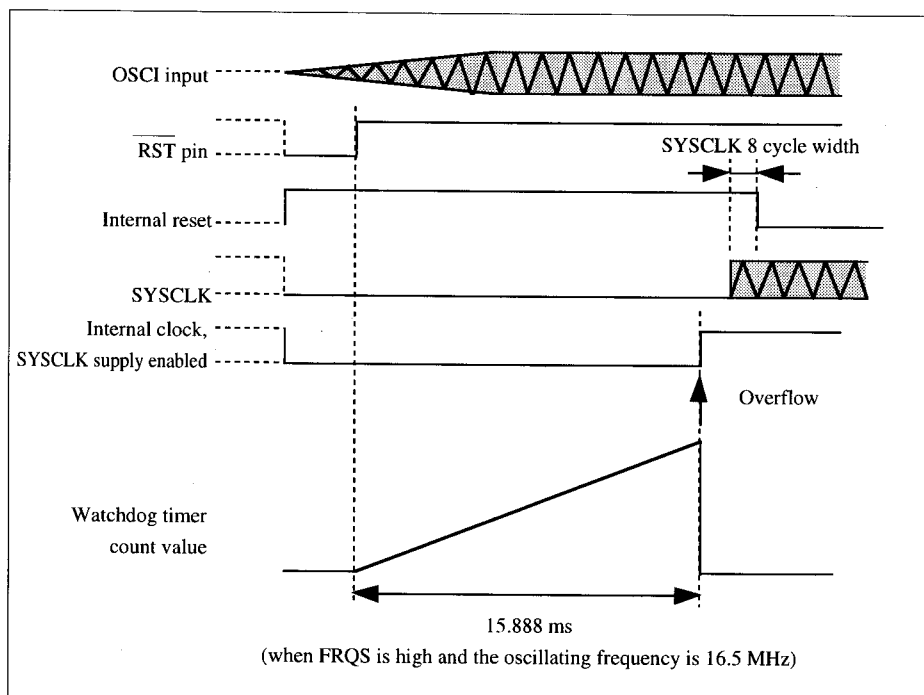
Fig. 10-5-1 Operation Diagram 1: When Reset Is Released
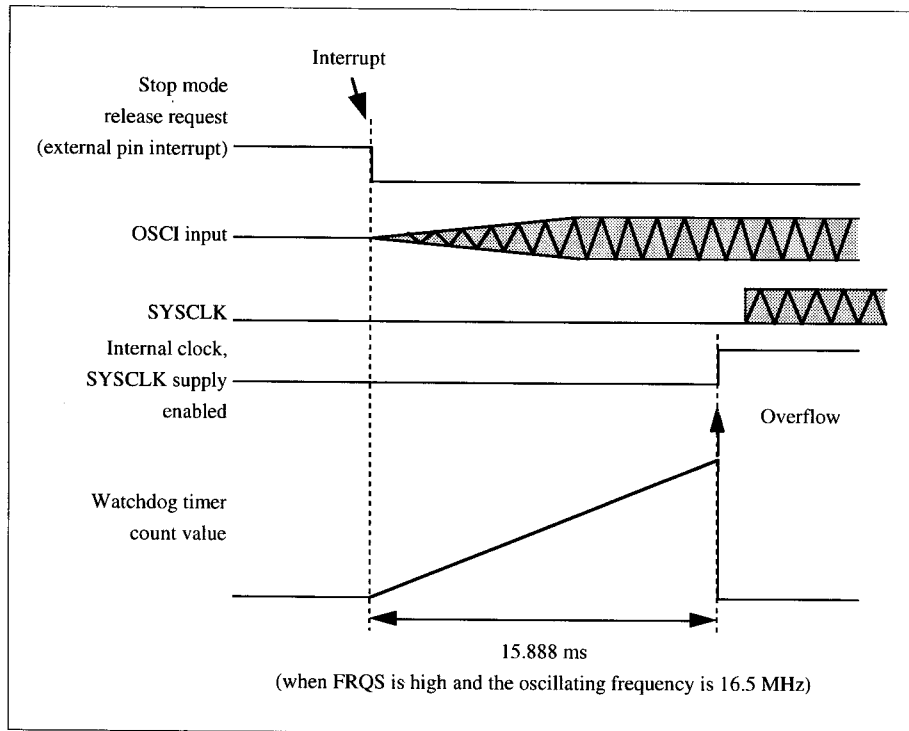
Fig. 10-5-2 Operation Diagram 2: When Recovering from Stop Mode

10.5.2 Watchdog Operation

If the WDCNE flag is set to "1" and the watchdog operation is enabled, a nonmaskable interrupt is generated if a watchdog timer overflow occurs.

When an overflow occurs, the watchdog timer overflow output is output to the WDOVF pin. Pulse output or level output can be selected through the WDOVT flag. The value output on the WDOVF pin can be read.

The watchdog timer overflow output is cleared by writing a "1" to the WDRST flag or by reset pin input.
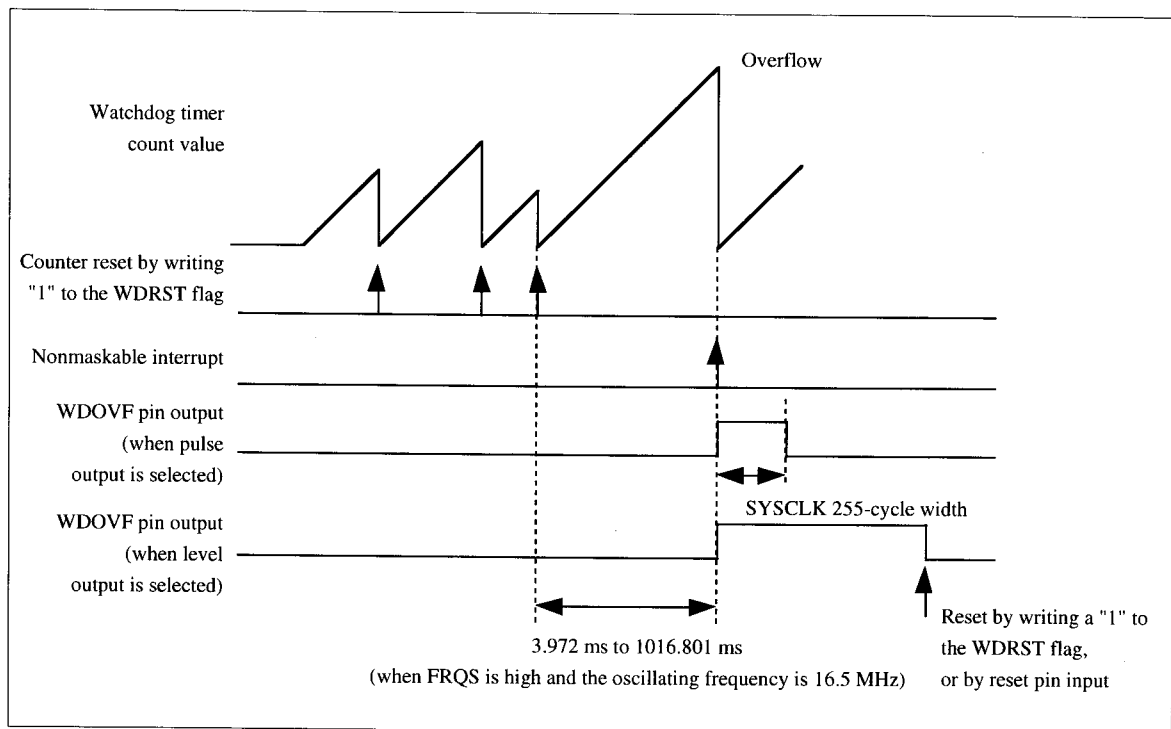
Fig. 10-5-3 Operation Diagram 3: Watchdog Operation

Before setting the WDCNE flag to "1", write a "1" to the WDRST flag to reset the counter.

When switching to HALT or SLEEP mode, set the WDCNE flag to "0" to turn off the watchdog timer.

10.5.3 Self-reset Operation

The chip resets internally when a "1" is written to the CHIPRST bit in the RSTCTR register. The oscillation stabilization wait operation is not performed.

The reset generated by writing the CHIPRST flag is an internal reset signal within the chip and does not manifest itself on the external reset pin.

Chapter 11.   Serial Interfaces

11

11.1 Overview

This microcontroller has two types of serial interfaces built in. One type has two channels (serial interfaces 0 and 1) and permits specification of start-stop synchronous mode, clock synchronous mode, and I2C mode. The other type has one channel (serial interface 2) and is a start-stop synchronous-only interface.
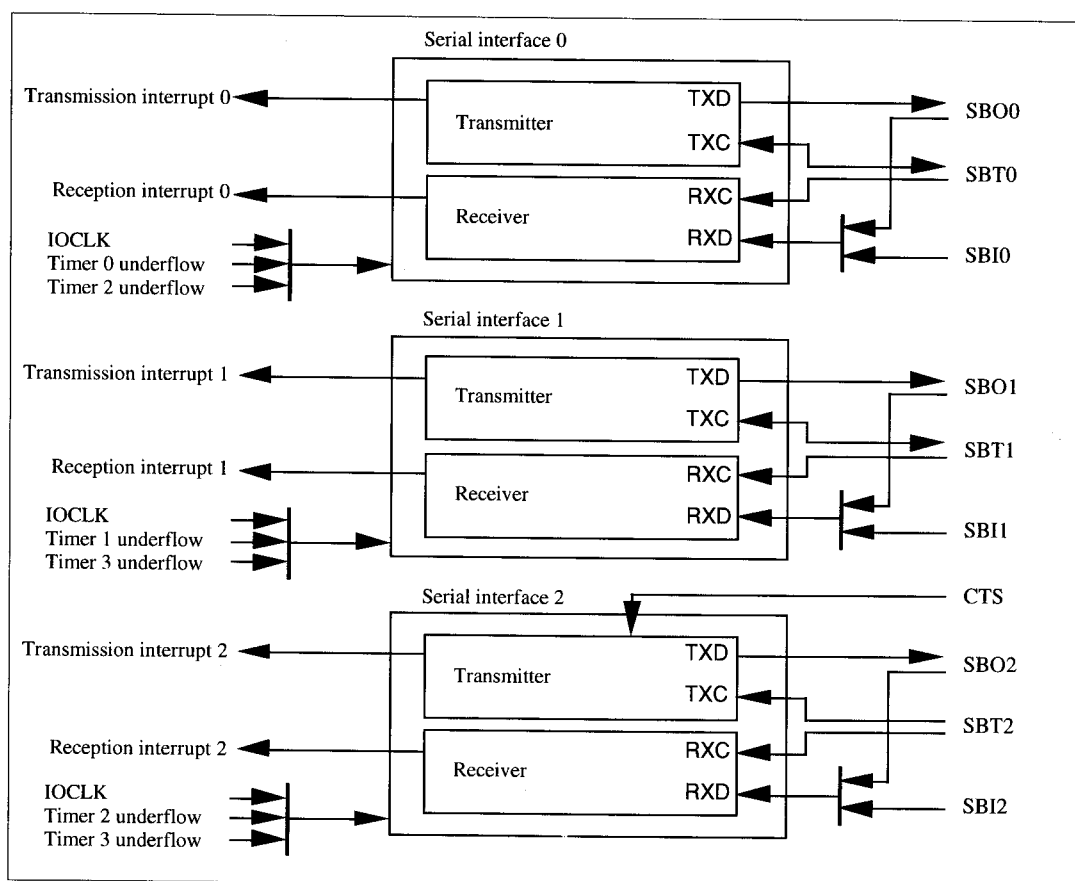
Fig. 11-1-1 Structural Diagram

11.2 Features

11.2.1 Serial Interface 0 and 1

<Clock synchronous mode>

• Parity	None, 0 fixed, 1 fixed, even, odd
• Character length	7 bits, 8 bits
• Transmission bit sequence	LSB or MSB selectable
• Clock source	1/2, 1/8, or 1/32 of IOCLK
	1/8 of timer 0 (timer 1) or timer 2 (timer 3) underflow,
	or 1/2 of timer 2 (timer3) underflow,
	External clock
• Maximum transfer speed	7.5 Mbps (when IOCLK is 15 MHz)
• Error detection during reception	Parity errors, overrun errors
• Buffers	Independent buffers for transmission and reception
	Reception and transmission buffers are both double buffers
• Interrupts	Transmission interrupts: "Transmission end" or "transmission buffer empty" selectable
	Reception interrupts: "Reception end" or "reception end with error" selectable
• DMA requests	During transmission: "Transmission end" or "transmission buffer empty" selectable
	During reception: Reception end

<Start-stop synchronous mode>

• Parity	None, 0 fixed, 1 fixed, even, odd
• Character length	7 bits, 8 bits
• Transmission bit sequence	LSB or MSB selectable
• Clock source	1/8 or 1/32 of IOCLK
	1/8 of timer 0 (timer 1) or timer 2 (timer 3) underflow
	1/8 of external clock
• Maximum transfer speed	19.2 kbps (when IOCLK is 15 MHz)
• Error detection during reception	Parity errors, overrun errors, framing errors
• Buffers	Independent buffers for transmission and reception
	Reception and transmission buffers are both double buffers
• Interrupts	Transmission interrupts: "Transmission end" or "transmission buffer empty" selectable
	Reception interrupts: "Reception end" or "reception end with error" selectable
• DMA requests	During transmission: "Transmission end" or "transmission buffer empty" selectable
	During reception: Reception end

<I2C mode>

• Master transmission, master reception possible (No start sequence conflict detection function)

11.2.2 Serial Interface 2

• Parity	None, 0 fixed, 1 fixed, even, odd
• Character length	7 bits, 8 bits
• Transmission bit sequence	LSB or MSB selectable
• Clock source	IOCLK
	Timer 2 or timer 3 underflow,
	External clock
• Maximum transfer speed	115.2 kbps (when IOCLK is 15 MHz)
• Error detection during reception	Parity errors, overrun errors, framing errors
• Transmission interrupt	Transmission operation can be interrupted through the CTS pin
• Buffers	Independent buffers for transmission and reception
	Reception and transmission buffers are both double buffers
• Interrupts	Transmission interrupts: "Transmission end" or "transmission buffer empty" selectable
	Reception interrupts: "Reception end" or "reception end with error" selectable
• DMA requests	During transmission: "Transmission end" or "transmission buffer empty" selectable
	During reception: Reception end

11.3 Block Diagram

11.3.1 Serial Interface 0 and 1
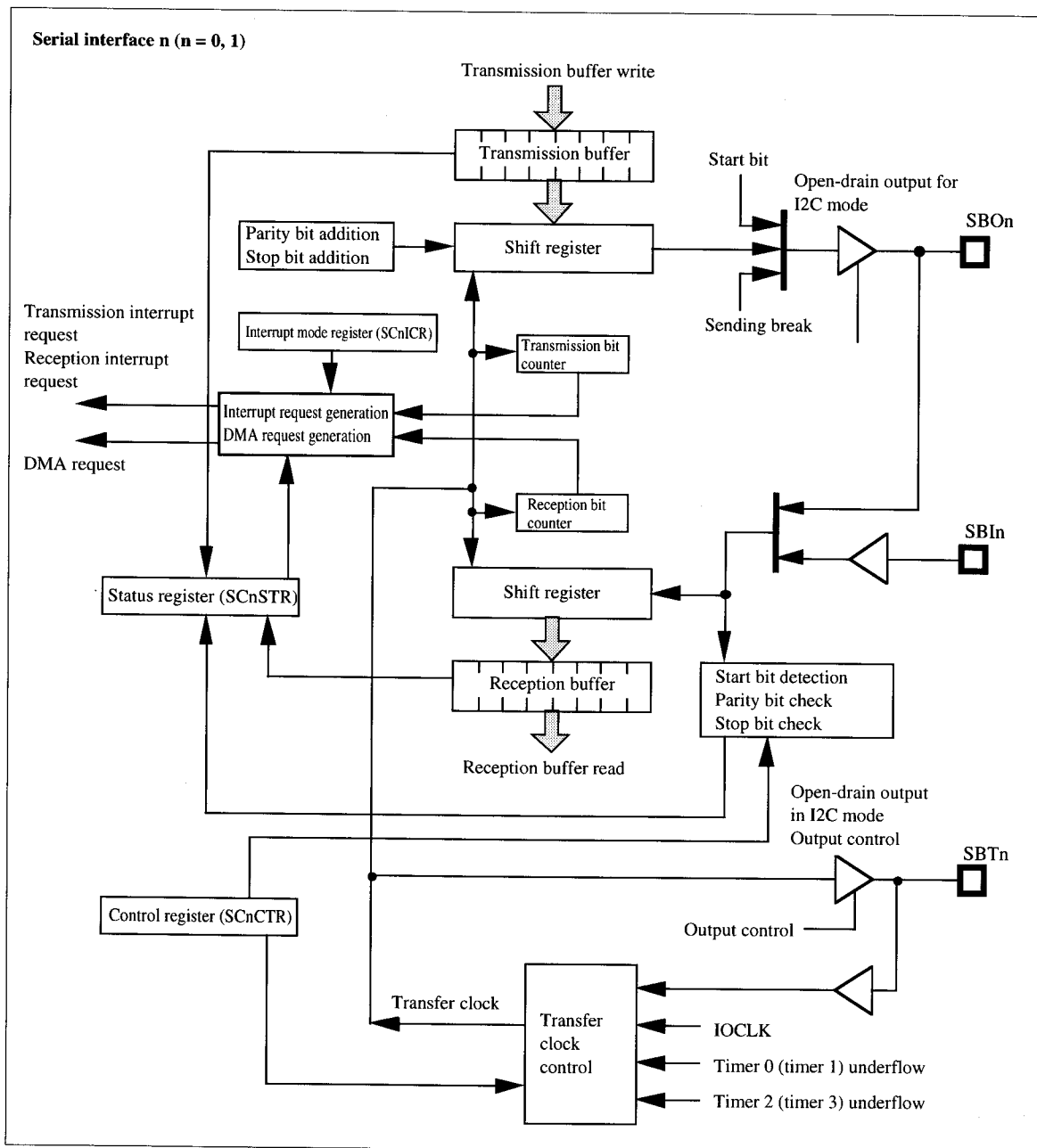
Fig. 11-3-1 Serial Interface 0 (Serial Interface 1) Block Diagram

11.3.2 Serial Interface 2
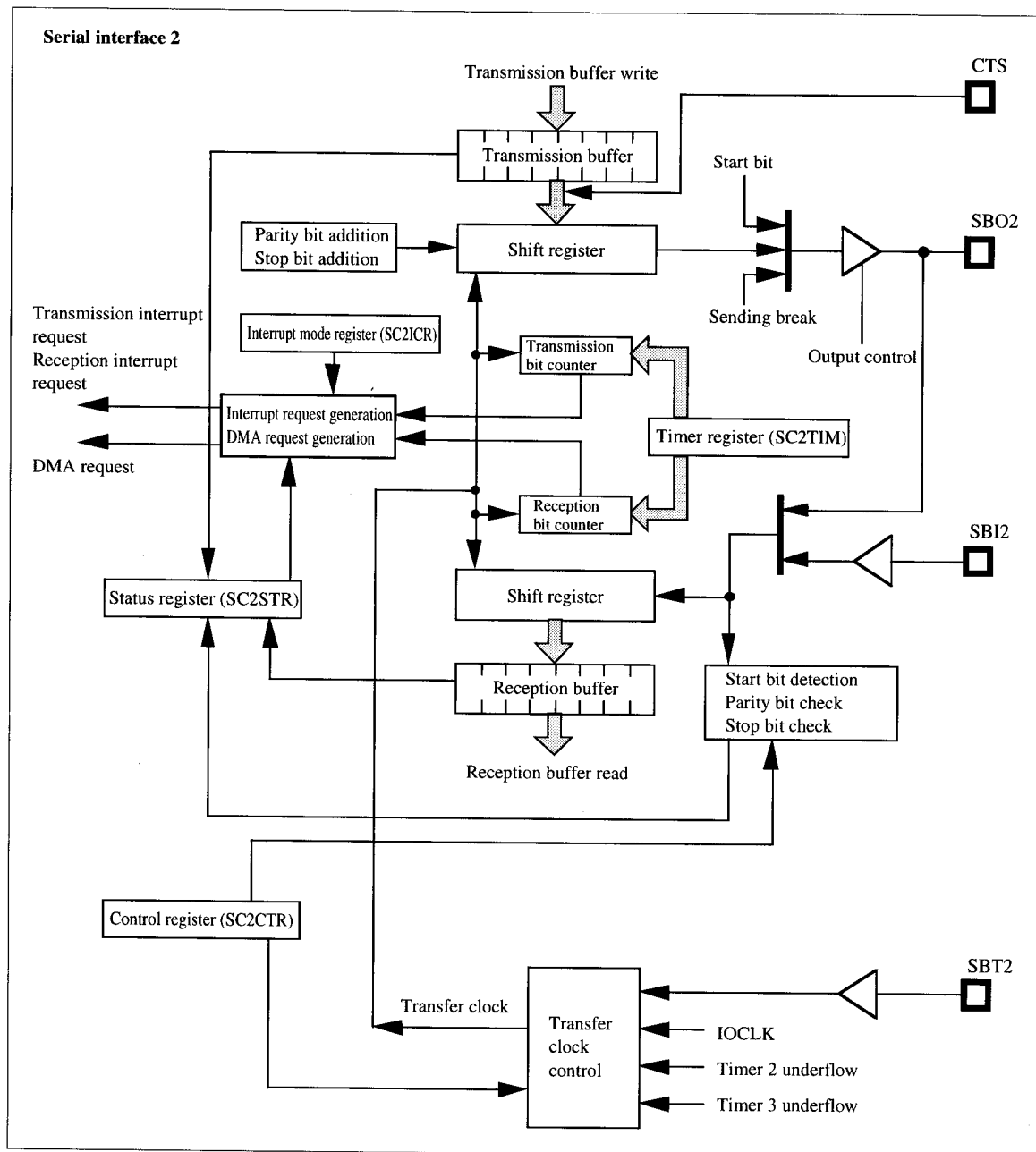
Fig. 11-3-2 Serial Interface 2 Block Diagram

11.4 Description of Registers

Table 11-4-1 List of Serial Interface Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'34000800	Serial 0 control register	SC0CTR	16	x'0000	8, 16
x'34000804	Serial 0 interrupt mode register	SC0ICR	8	x'00	8
x'34000808	Serial 0 transmission buffer	SC0TXB	8	x'00	8
x'34000809	Serial 0 reception buffer	SC0RXB	8	x'00	8
x'3400080C	Serial 0 status register	SC0STR	16	x'0000	8, 16
x'34000810	Serial 1 control register	SC1CTR	16	x'0000	8, 16
x'34000814	Serial 1 interrupt mode register	SC1ICR	8	x'00	8
x'34000818	Serial 1 transmission buffer	SC1TXB	8	x'00	8
x'34000819	Serial 1 reception buffer	SC1RXB	8	x'00	8
x'3400081C	Serial 1 status register	SC1STR	16	x'0000	8, 16
x'34000820	Serial 2 control register	SC2CTR	16	x'0000	8, 16
x'34000824	Serial 2 interrupt mode register	SC2ICR	8	x'00	8
x'34000828	Serial 2 transmission buffer	SC2TXB	8	x'00	8
x'34000829	Serial 2 reception buffer	SC2RXB	8	x'00	8
x'3400082C	Serial 2 status register	SC2STR	8	x'XX	8
x'3400082D	Serial 2 timer register	SC2TIM	8	x'00	8

* When writing to the serial interface registers, first set the I/O bus mode to synchronous mode. Operation is not guaranteed if the write is performed in asynchronous mode.

11.4.1 Serial n Control Register (n = 0, 1)

Register symbol: SCnCTR

Address: x' 34000800 (n=0), x' 34000810 (n=1)

Purpose: This register sets the serial interface n operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	SCn TXE	SCn RXE	SCn BKE	SCn IIC	SCn MD1	SCn MD0	SCn OD	SCn TOE	SCn CLN	SCn PB2	SCn PB1	SCn PB0	SCn STB	SCn CK2	SCn CK1	SCn CK0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	SCnCK0	Clock source selection (LSB)
1	SCnCK1	Clock source selection
2	SCnCK2	Clock source selection (MSB)

000:    1/2 IOCLK (valid only in clock synchronous mode)

001:    1/8 IOCLK

010:    1/32 IOCLK

011:    1/2 timer 2 (timer 3) underflow (valid only in clock synchronous mode)

100:    1/8 timer 0 (timer 1) underflow

101:    1/8 timer 2 (timer 3) underflow

110:    1/8 external clock (valid only in start-stop synchronous mode)

111:    External clock (valid only in clock synchronous mode)

| 3 | SCnSTB | Stop bit selection (valid only in start-stop synchronous mode) |

0: 1 bit   1: 2 bits

4	SCnPB0	Parity bit selection (LSB)
5	SCnPB1	Parity bit selection
6	SCnPB2	Parity bit selection (MSB)

000:    None

001, 010, 011: Setting prohibited

100:    0 fixed

101:    1 fixed

110:    Even (even number of 1's)

111:    Odd (odd number of 1's)

| 7 | SCnCLN | Character length selection |

0: 7 bits   1: 8 bits

<Continued>

<Continued>

Bit No.	Bit name	Description
8	SCnTOE	SBTn pin output control
		0: When the internal clock is selected, the SBTn pin is an output only while transmission is in progress (the SBTn pin is an input when in standby mode or when an external clock is selected)
		1: When the internal clock is selected, the SBTn pin is always an output (the SBTn pin is an input when an external clock is selected)
9	SCnOD	Transmission/reception bit sequence selection
		0: From LSB 1: From MSB
10	SCnMD0	Protocol selection (LSB)
11	SCnMD1	Protocol selection (MSB)
		00: Start-stop synchronous mode
		01: Clock synchronous mode (1)
		(the SBOn pin is used as a data output, and the SBIn pin is used as a data input)
		10: I2C mode
		11: Clock synchronous mode (2)
		(the SBOn pin is used as a data input and output, and input on the SBIn pin is ignored)
12	SCnIIC	I2C mode selection
		0: The stop sequence is output when this bit is changed from "1" to "0"
		1: The start sequence is output when this bit is changed from "0" to "1"
13	SCnBKE	Sending break
		0: Do not send break
		1: Send break (The SBOn pin is fixed to output "0".)
14	SCnRXE	Reception operation enable
		0: Disabled 1: Enabled
15	SCnTXE	Transmission operation enable
		0: Disabled 1: Enabled

11.4.2 Serial n Interrupt Mode Register (n = 0, 1)

Register symbol: SCnICR

Address: x' 34000804 (n=0), x' 34000814 (n=1)

Purpose: This register selects the sources for transmission interrupts, reception interrupts, and DMA startup requests for serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn DMD	-	SCn TD	SCn TI	-	SCn RES	-	SCn RI
When reset	0	0	0	0	0	0	0	0
Access	R/W	R	R/W	R/W	R	R/W	R	R/W

Bit No.	Bit name	Description
0	SCnRI	Reception interrupt source selection
		0: Reception end
		1: Reception end with error
1	—	Always returns "0".
2	SCnRES	Reception error interrupt source selection
		0: Interrupt request when an overrun, parity, or framing error occurs
		1: Interrupt request when a parity error occurs
3	—	Always returns "0".
4	SCnTI	Transmission interrupt source selection
		0: Transmission end 1: Transmission buffer empty
5	SCnTD	DMA startup source selection for transmission
		0: Transmission end 1: Transmission buffer empty
6	—	Always returns "0".

7

SCnDMD

Data output maintenance during external clock transmission

(valid only in clock synchronous mode)

0: Set data pin high at end of transmission

1: Maintain data pin at end of transmission

11.4.3 Serial n Transmission Buffer (n = 0, 1)

Register symbol: SCnTXB

Address: x'34000808 (n=0) , x'34000818 (n=1)

Purpose: This register writes the transmission data to serial interface n.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn TXB7	SCn TXB6	SCn TXB5	SCn TXB4	SCn TXB3	SCn TXB2	SCn TXB1	SCn TXB0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data is transmitted by writing it to this buffer.

11.4.4 Serial n Reception Buffer (n = 0, 1)

Register symbol: SCnRXB

Address: x'34000809 (n=0) , x'34000819 (n=1)

Purpose: This register reads in the reception data.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SCn RXB7	SCn RXB6	SCn RXB5	SCn RXB4	SCn RXB3	SCn RXB2	SCn RXB1	SCn RXB0
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Reception data is gotten by reading this buffer at the end of reception. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

11.4.5 Serial n Status Register (n = 0, 1)

Register symbol: SCnSTR

Address: x'3400080C (n=0), x'3400081C (n=1)

Purpose: This register indicates the status of serial interface n.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	SCn SPF	SCn STF	SCn TXF	SCn RXF	SCn TBF	SCn RBF	-	SCn FEF	SCn PEF	SCn OEF
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit No.	Bit name	Description
0	SCnOEF	Overrun error indication

0: No error

1: Overrun error occurred

1	SCnPEF	Parity error indication

0: No error

1: Parity error occurred

2	SCnFEF	Framing error indication

0: No error

1: Framing error occurred

3 — Always returns "0".

4	SCnRBF	Reception buffer status indication

0: Reception buffer empty

1: Data exists in the reception buffer

5	SCnTBF	Transmission buffer status indication

0: Transmission buffer empty

1: Data exists in the transmission buffer

6	SCnRXF	Reception status indication

0: Waiting for reception

1: Reception in progress

7	SCnTXF	Transmission status indication

0: Ready for transmission

1: Transmission in progress

8	SCnSTF	I2C start sequence detection

(Cleared by reading SCnRXB or by writing SCnTXB)

0: Not detected

1: Detected

9	SCnSPF	I2C stop sequence detection

(Cleared by reading SCnRXB or by writing SCnTXB)

0: Not detected

1: Detected

10 to 15 — Always returns "0".

11.4.6 Serial 2 Control Register

Register symbol: SC2CTR

Address: x'34000820

Purpose: This register sets the serial interface 2 operation control conditions.

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	SC2 TXE	SC2 RXE	SC2 BKE	SC2 TWS	-	-	SC2 OD	SC2 TWE	SC2 CLN	SC2 PB2	SC2 PB1	SC2 PB0	SC2 STB	-	SC2 CK1	SC2 CK0
When reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit No.	Bit name	Description
0	SC2CK0	Clock source selection (LSB)
1	SC2CK1	Clock source selection (MSB)

00: IOCLK

01: Timer 2 underflow

10: External clock

11: Timer 3 underflow

2	—	Always returns "0".
3	SC2STB	Stop bit selection

0: 1 bit   1: 2 bits

4	SC2PB0	Parity bit selection (LSB)
5	SC2PB1	Parity bit selection
6	SC2PB2	Parity bit selection (MSB)

000: None

001, 010, 011: Setting prohibited

100: 0 fixed

101: 1 fixed

110: Even (even number of 1's)

111: Odd (odd number of 1's)

7	SC2CLN	Character length selection

0: 7 bits   1: 8 bits

8	SC2TWE	Transmission interruption enable

0: Disabled   1: Enabled

9	SC2OD	Transmission bit sequence selection

0: From LSB   1: From MSB

10 to 11	—	Always returns "0".

<Continued>

<Continued>

Bit No.	Bit name	Description
12	SC2TWS	Transmission interruption code selection 0: Interrupt when low 1: Interrupt when high
13	SC2BKE	Sending break 0: Do not send break 1: Send break (The SBOn pin is fixed to output "0".)
14	SC2RXE	Reception operation enable 0: Disabled 1: Enabled
15	SC2TXE	Transmission operation enable 0: Disabled 1: Enabled

11.4.7 Serial 2 Interrupt Mode Register

Register symbol:	SC2ICR
Address:	x'34000824 (n=0)
Purpose:	This register selects the sources for transmission interrupts, reception interrupts, and DMA startup requests for serial interface 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	SC2 TD	SC2 TI	-	SC2 RES	-	SC2 RI
When reset	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R	R/W	R	R/W

Bit No.	Bit name	Description
0	SC2RI	Reception interrupt source selection 0: Reception end 1: Reception end with error
1	—	Always returns "0".
2	SC2RES	Reception error interrupt source selection 0: Interrupt request when an overrun, parity, or framing error occurs 1: Interrupt request when a parity error occurs
3	—	Always returns "0".
4	SC2TI	Transmission interrupt source selection 0: Transmission end 1: Transmission buffer empty
5	SC2TD	DMA startup source selection for transmission 0: Transmission end 1: Transmission buffer empty
6 to 7	—	Always returns "0".

11.4.8   Serial 2 Transmission Buffer

Register symbol: SC2TXB

Address: x'34000828

Purpose: This register writes the transmission data to serial interface 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC2 TXB7	SC2 TXB6	SC2 TXB5	SC2 TXB4	SC2 TXB3	SC2 TXB2	SC2 TXB1	SC2 TXB0
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data is transmitted by writing it to this buffer.

11.4.9   Serial 2 Reception Buffer

Register symbol: SC2RXB

Address: x'34000829

Purpose: This register reads in the reception data.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC2 RXB7	SC2 RXB6	SC2 RXB5	SC2 RXB4	SC2 RXB3	SC2 RXB2	SC2 RXB1	SC2 RXB0
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Reception data is gotten by reading this buffer at the end of reception. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

11.4.10 Serial 2 Status Register

Register symbol: SC2STR

Address: x'3400082C

Purpose: This register indicates the status of serial interface 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	SC2 TXF	SC2 RXF	SC2 TBF	SC2 RBF	SC2 CTS	SC2 FEF	SC2 PEF	SC2 OEF
When reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

Bit No.	Bit name	Description
0	SC2OEF	Overrun error indication
		0: No error
		1: Overrun error occurred
1	SC2PEF	Parity error indication
		0: No error
		1: Parity error occurred
2	SC2FEF	Framing error indication
		0: No error
		1: Framing error occurred
3	SC2CTS	CTS status indication
		0: CTS pin is low
		1: CTS pin is high
4	SC2RBF	Reception buffer status indication
		0: Reception buffer empty
		1: Data exists in the reception buffer
5	SC2TBF	Transmission buffer status indication
		0: Transmission buffer empty
		1: Data exists in the transmission buffer
6	SC2RXF	Reception status indication
		0: Waiting for reception
		1: Reception in progress
7	SC2TXF	Transmission status indication
		0: Ready for transmission
		1: Transmission in progress

11.4.11 Serial 2 Timer Setting Register

Register symbol: SC2TIM

Address: x'3400082D

Purpose: This register sets the internal timer for serial interface 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	SC2 TIM6	SC2 TIM5	SC2 TIM4	SC2 TIM3	SC2 TIM2	SC2 TIM1	SC2 TIM0
When reset	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The cycle is the value set in the register - 1.

11.5 Description of Operation

11.5.1 Serial Interfaces 0 and 1

11.5.1.1 Connection

<Clock synchronous mode>

Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.

When the SBTn pin is an output only during transmission (SCnTOE = "0"), it is necessary to pull up the SBTn pin. In addition, when using the SBOn pin as a data input/output (SCnMD1 and 0 = "11"), it is necessary to pull up the SBOn pin. Connect pull-up resistors externally .

When using the SBOn pin as a data output and the SBIn pin as a data input (SCnMD1 and 0 = "01"), the SBOn pin is always an output and the SBIn pin is always an input.

When using the SBOn pin as a data input/output (SCnMD1 and 0 = "11"), the SBOn pin is an output only during transmission, and is normally an input.

When SCnTOE is "0", the SBTn pin is an output only during transmission under the internal clock, and is normally an input. Furthermore, when SCnTOE is "1", the SBTn pin is always an output when the internal clock is selected.
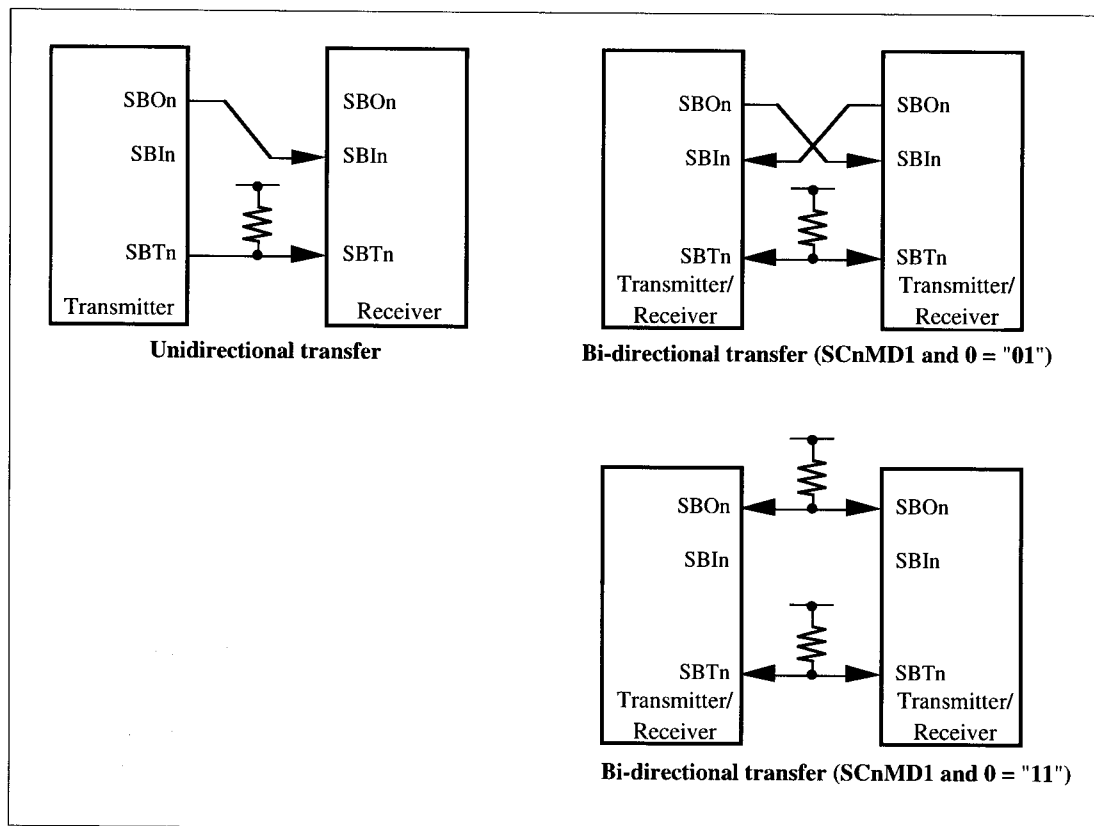
Fig. 11-5-1 Connections (1)

<Start-stop synchronous mode>

Two different connection methods are possible, one for unidirectional transfer, and the other for bi-directional transfer.

The SBOn pin is always an output, and the SBIn pin is always an input.
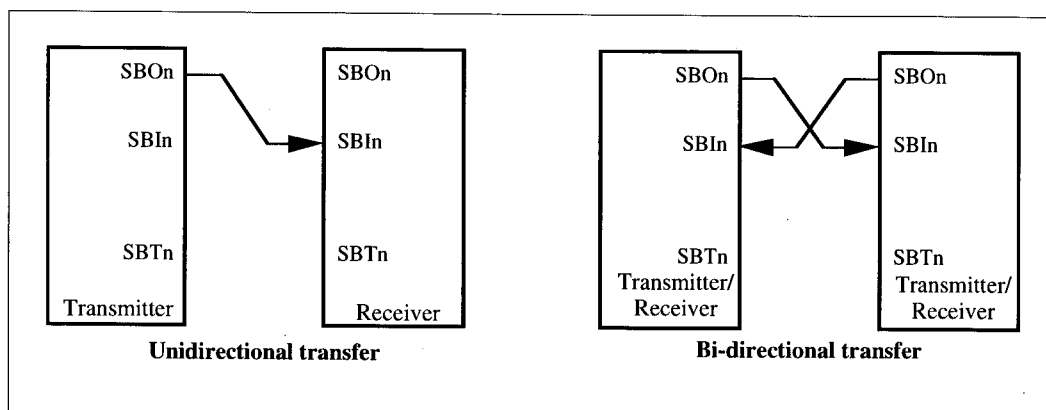
Fig. 11-5-2 Connections (2)

<I2C mode>

It is possible to connect devices that are capable of slave transmission and slave reception.

SDA and SCL require pull-up resistors. Connect pull-up resistors externally.

The SBOn pin is an open-drain input/output, and the SBTn pin is an open drain output.
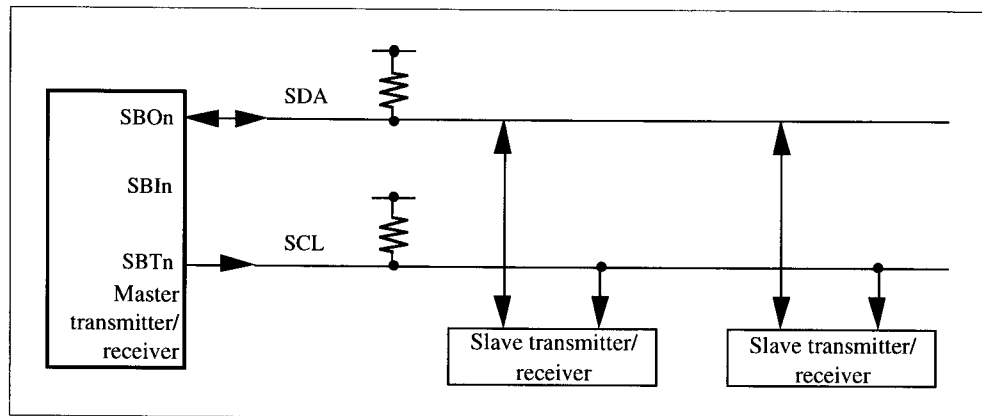
Fig. 11-5-3 Connections (3)

11.5.1.2 Baud rates

<Start-stop synchronous mode>

Select a suitable baud rate and serial interface input clock in start-stop synchronous mode.
Set the input clock as follows:

Input clock = Desired baud rate × 8

When using IOCLK, divide it with transfer timers 0 and 2 (timers 1 and 3). Set the division ratio as follows:

Timer division ratio = ABS(IOCLK frequency/desired baud rate / 8 + 0.5)

When the division rate is large, either use timer 0 as a prescaler, or else cascade timers 1 and 2 (timers 2 and 3).
When using IOCLK as the input clock, the baud rate that can actually be set is as follows:

Baud rate = IOCLK frequency/timer division ratio/8

In this case, the baud rate error is determined according to the following formula:

Baud rate error = ABS(IOCLK frequency/timer division ratio/8/desired baud rate - 1)

Examples using IOCLK are shown below.

Table 11-5-1 Baud Rates (1) (when IOCLK = 15 MHz)

Baud rate (bps)	When cascaded		When using prescalers	
	Timer division ratio	Transfer rate error	Timer division ratio	Transfer rate error
19 200	98*	0.35 %	98*	0.35 %
9 600	195*	0.16 %	195*	0.16 %
4 800	391	0.10 %	390 (195 ×2)	0.16 %
2 400	781	0.03 %	780 (195 ×4)	0.16 %
1 200	1 563	0.03 %	1 560 (195 × 8)	0.16 %

Table 11-5-2 Baud Rates (2) (when IOCLK = 12 MHz)

Baud rate (bps)	When cascaded		When using prescalers	
	Timer division ratio	Transfer rate error	Timer division ratio	Transfer rate error
19 200	78*	0.16 %	78*	0.35 %
9 600	156*	0.16 %	156*	0.16 %
4 800	313	0.16 %	312 (156 × 2)	0.16 %
2 400	625	0.00 %	625 (125 × 4)	0.00 %
1 200	1 250	0.00 %	1 250 (125 × 8)	0.00 %

Table 11-5-3 Baud Rates (3) (when IOCLK = 10 MHz)

Baud rate (bps)	When cascaded		When using prescalers	
	Timer division ratio	Transfer rate error	Timer division ratio	Transfer rate error
19 200	65*	0.16 %	65*	0.16 %
9 600	130*	0.16 %	130*	0.16 %
4 800	260	0.16 %	260 (130 × 2)	0.16 %
2 400	521	0.03 %	520 (130 × 4)	0.16 %
1 200	1 042	0.03 %	1 040 (130 × 8)	0.16 %

* : It is not necessary to use cascaded connection or prescaler.

When using 1/8 of an external clock as the clock source, the widths of the high and low pulses of the input clock must be at least two IOCLK cycles.

11.5.1.3 Clock Synchronous Mode Timing

<Transmission>

• One-byte transfer with 8-bit data length and parity on
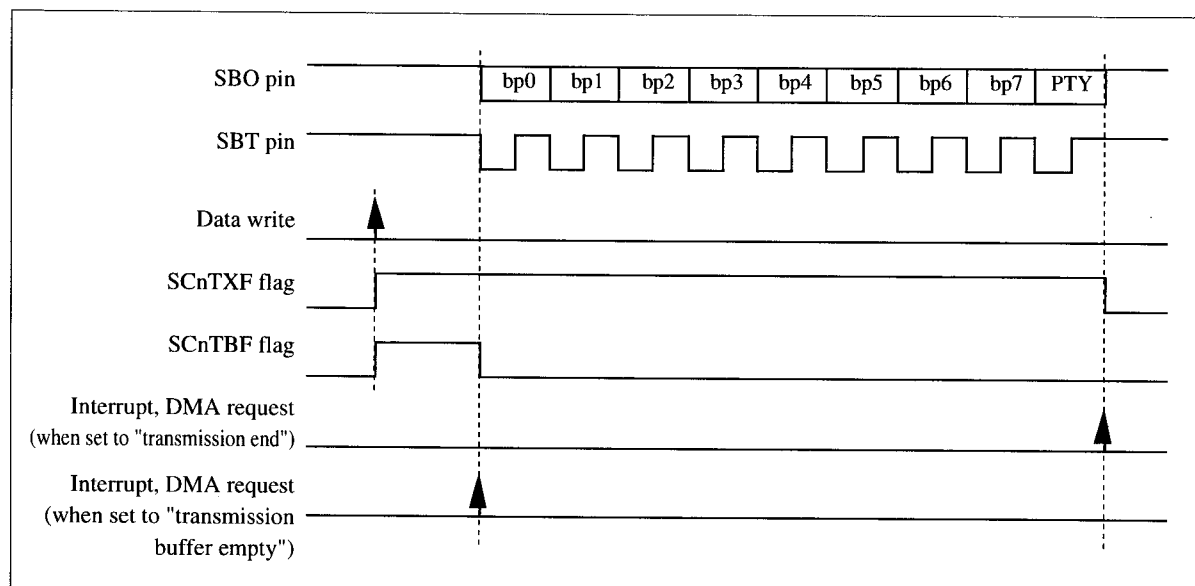
Fig. 11-5-4 Timing Chart (1)

• Two-byte transfer with 8-bit data length and parity off
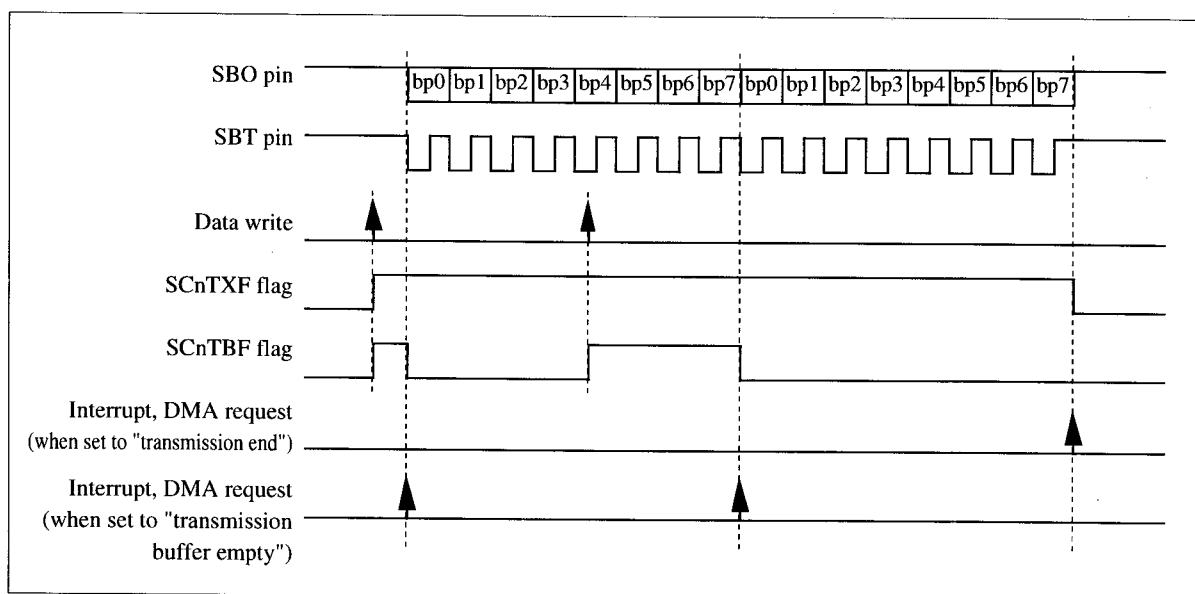
Fig. 11-5-5 Timing Chart (2)

When transmission is enabled, transmission starts when data is written to SCnTXB.

Continuous transmission is possible by writing data to SCnTXB again while transmission is in progress.

During a 7-bit transfer, the MSB (bit 7) is ignored.

The SCnTXF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the end of transmission.

The SCnTBF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the start of transmission.

<Reception>

• One-byte transfer with 8-bit data length and parity on
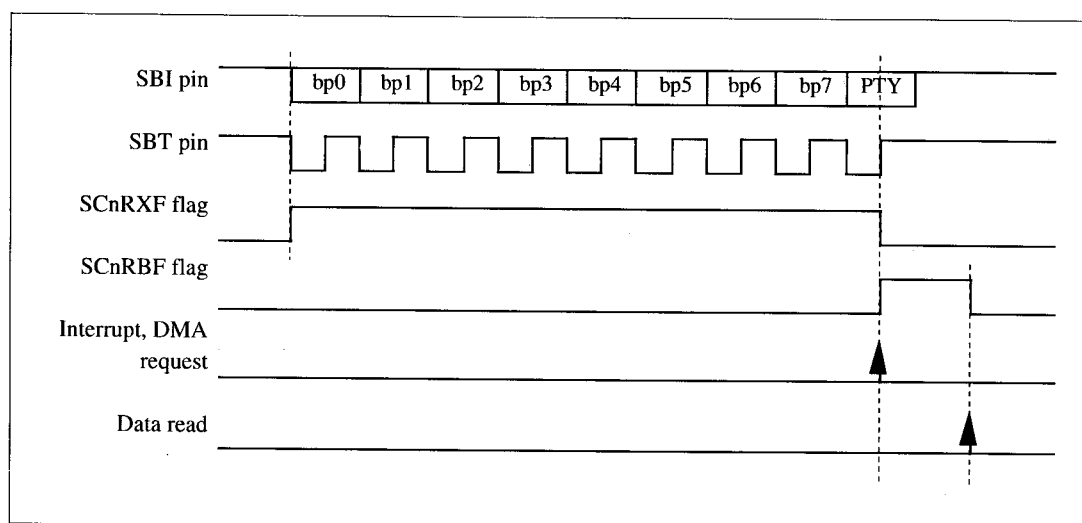
Fig. 11-5-6 Timing Chart (3)

• Two-byte transfer with 8-bit data length and parity off
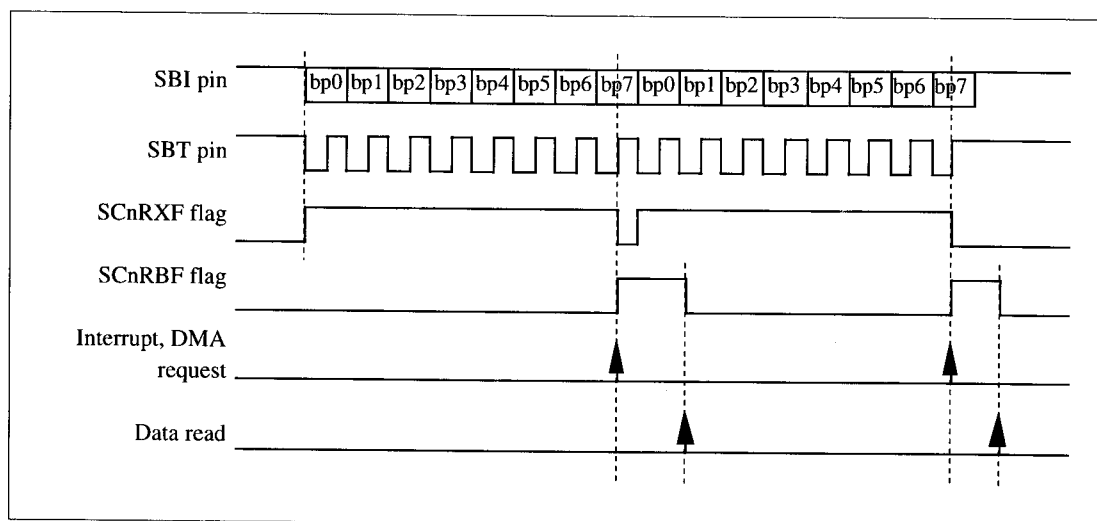
Fig. 11-5-7 Timing Chart (4)

After reception end (when the SCnRBF flag is "1"), the received data is fetched by reading the SCnRXB register. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

The SCnRXF flag is set to "1" at the start of reception (at the falling edge of SBT), and is set to "0" at the end of reception.
The SCnRBF flag is set to "1" at the end of reception, and is set to "0" when SCnRXB is read.

An overrun error is generated when reception of the next data is completed before previously received data has been read from the SCnRXB register. In this event, the previously received data is lost. The overrun error indicator flag (SCnOEF) is updated at the moment the final data bit is received.

A parity error is generated when the parity bit is fixed to "0" and a "1" is received, when the parity bit is fixed to "1" and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SCnPEF) is updated at the moment the parity bit is received.

11.5.1.4. Start-stop Synchronous Mode Timing

<Transmission>

- Transfer with 8-bit data length, parity on, and 1 stop bit
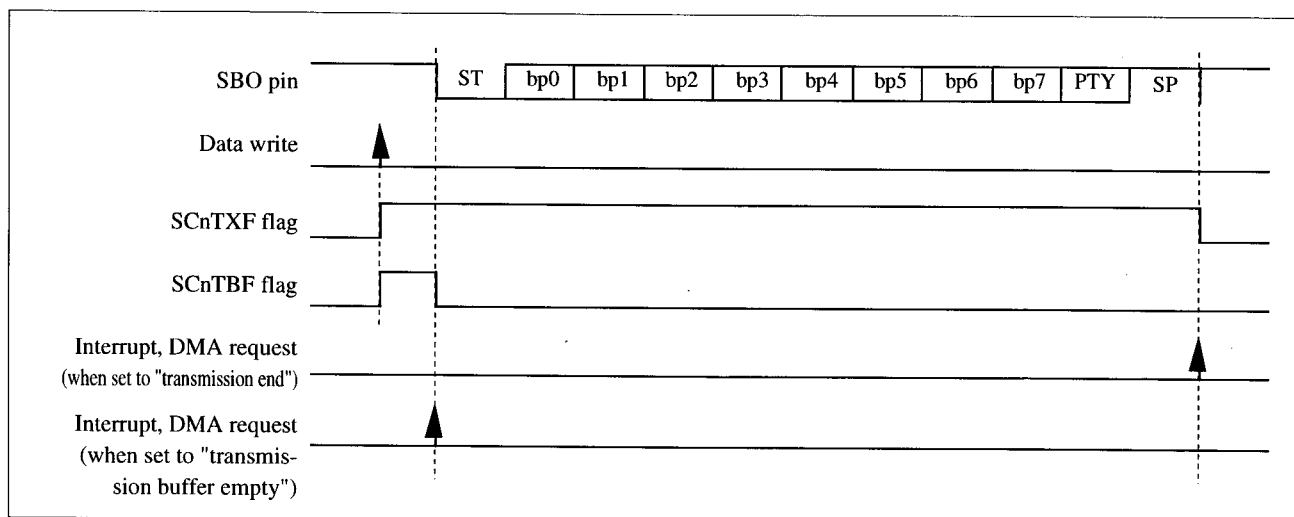
Fig. 11-5-8 Timing Chart (5)

- Two-byte transfer with 7-bit data length, parity on, and 1 stop bit
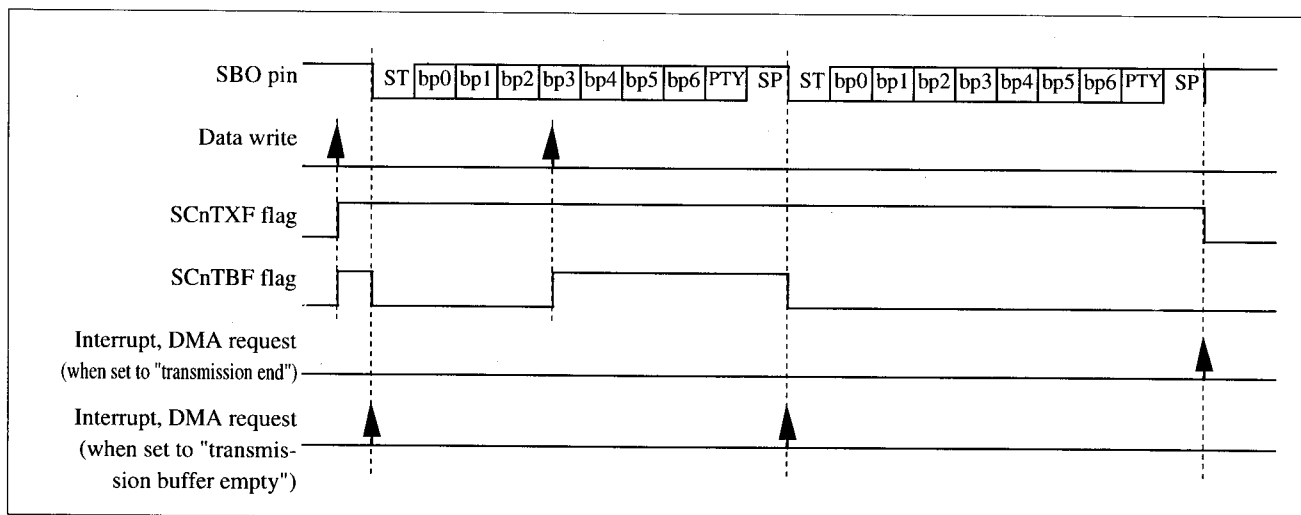
Fig. 11-5-9 Timing Chart (6)

When transmission is enabled, transmission starts when data is written to SCnTXB.

Continuous transmission is possible by writing data to SCnTXB again while transmission is in progress.

During a 7-bit transfer, the MSB (bit 7) is ignored.

The SCnTXF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the end of transmission.
The SCnTBF flag is set to "1" when data is written to SCnTXB, and is set to "0" at the start of transmission.

<Reception>

• Transfer with 8-bit data length, parity on, and 1 stop bit
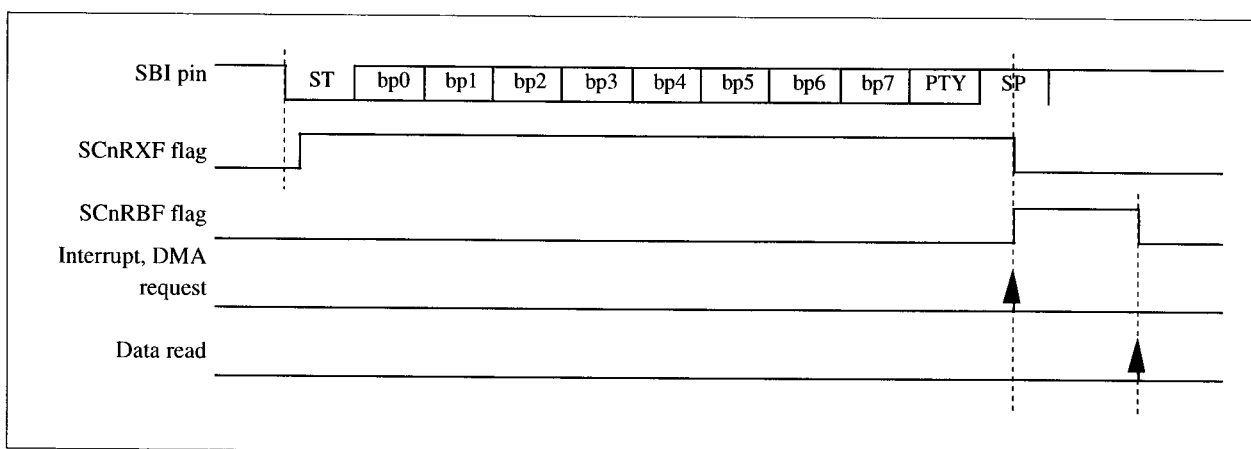
Fig. 11-5-10 Timing Chart (7)

• Two-byte transfer with 7-bit data length, parity on, and 1 stop bit
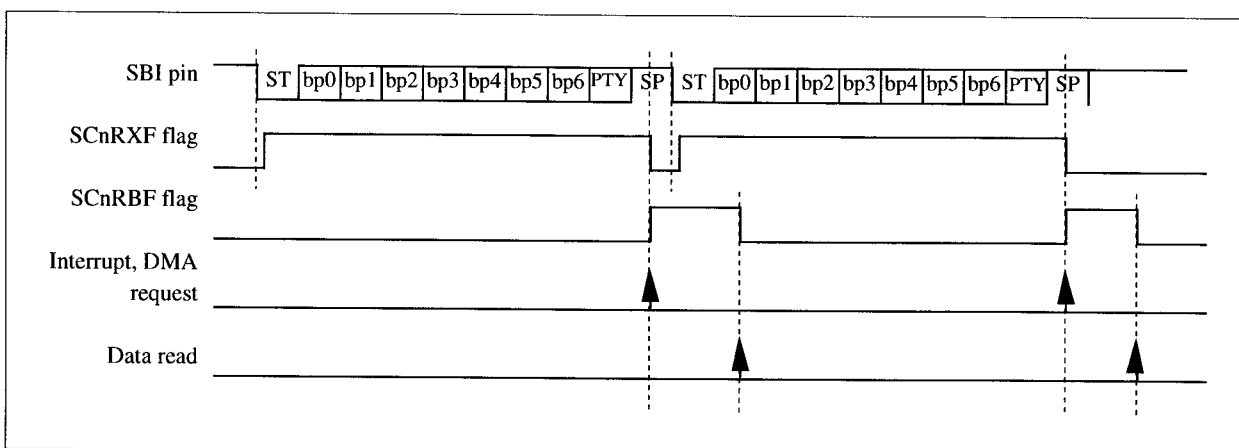
Fig. 11-5-11 Timing Chart (8)

After reception end (when the SCnRBF flag is "1"), the received data is fetched by reading the SCnRXB register. In the case of a 7-bit transfer, the MSB (bit 7) is "0".

The SCnRXF flag is set to "1" at the start of reception (when the start bit is detected), and is set to "0" at the end of reception.

The SCnRBF flag is set to "1" at the end of reception, and is set to "0" when SCnRXB is read.

An overrun error is generated when reception of the next data is completed before previously received data has been read from the SCnRXB register. In this event, the previously received data is lost. The overrun error indicator flag (SCnOEF) is updated at the moment the final data bit is received.

A parity error is generated when the parity bit is fixed to "0" and a "1" is received, when the parity bit is fixed to "1" and a "0" is received, when even parity is set and an odd number of ones is received, or when odd parity is set and an even number of ones is received. The parity error indicator flag (SCnPEF) is updated at the moment the parity bit is received.

A framing error is generated when "0" was received for the stop bit. The framing error indicator flag (SCnFEF) is updated at the moment the stop bit is received.

11.5.1.5 I2C Mode Timing

<Master transmission>
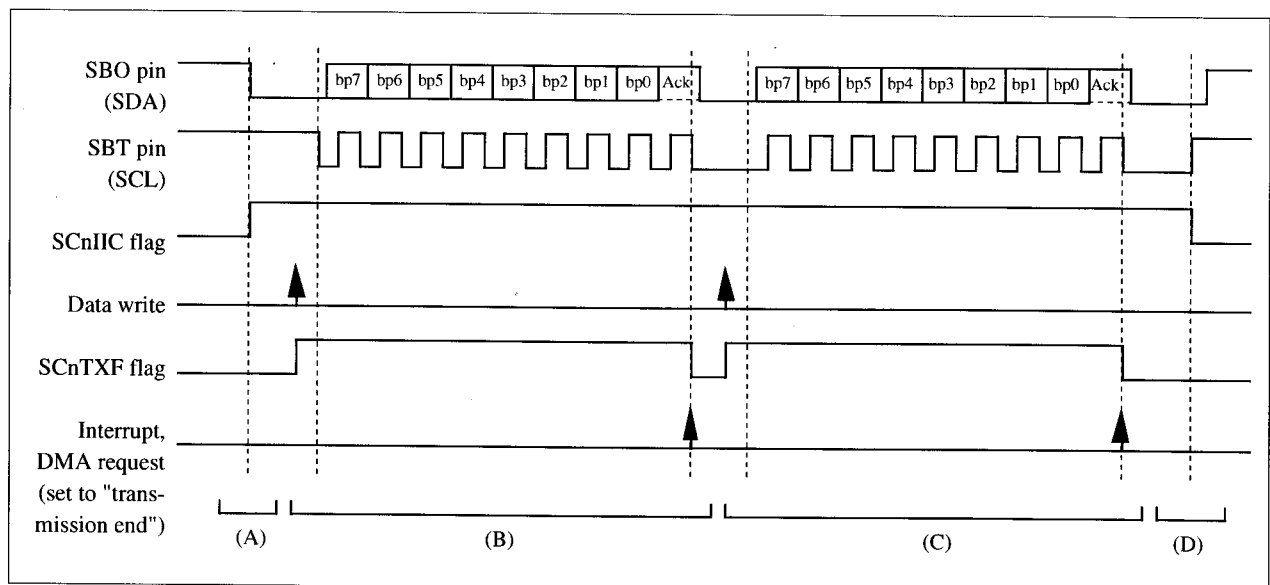
Fig. 11-5-12 Timing Chart (9)

Initial Settings

(1) Select the transmission clock. (SCnCK2 to 0)

  Setting an external clock is prohibited.

(2) The parity bit is used as a substitute for Ack. When transmitting, set the parity bit to "'1' fixed". (SCnPB2 to 0)

  When detecting an Ack signal output by the receiving slave device, enable the receiving operation, even when transmitting. (The signal is detected as a parity error.)

(3) Set the character length and the transmission bit sequence. (SCnCLN, SCnOD)

(4) Set the I2C mode selection flag (SCnIIC) to "0".

(5) Set the protocol to I2C mode, and set the SBT pin as an output only when transmission is in progress. (SCnMD1 and 0, SCnTOE)

  The SBO and SBT pins are open-drain outputs.

(6) Enable the transmission operation. (SCnTXE)

  Enable reception also when the Ack signal is detected and during master reception. (SCnRXE)

(A) Start sequence send

(1) When a "1" is written to the I2C mode selection flag (SCnIIC), a low signal is output on the SBO pin as the start sequence.

  When the start sequence is generated normally, the I2C start sequence detection bit (SCnSTF) is set to "1". In this case, even if there is another simultaneous start, the "arbitration lost" state is not detected.

(B) Data transmission 1

(1) Data is transmitted once the data is written to the serial transmission buffer (SCnTXB). The SBOn pin output changes after the falling edge of the SBTn pin signal.

(2) After the transmission ends, the SBOn pin output and the SBTn pin output are both maintained low.

(C) Data transmission 2

(1) When sending data continuously, write the data to the serial transmission buffer (SCnTXB).

(D) Stop bit send

(1) When ending the data transmission, write a "0" to the I2C mode selection flag (SCnIIC). Be sure to write a "0" to this flag only when transmission possible (flag SCnTXF is "0") and reception buffer is empty (flag SCnRBF is "0").

(2) Simultaneously with the write, the SBTn pin output goes high. After one cycle, the SBOn pin output goes high, and the stop sequence is sent. At this point, the I2C stop sequence detection bit (SCnSPF) is set to "1".

<Master reception>
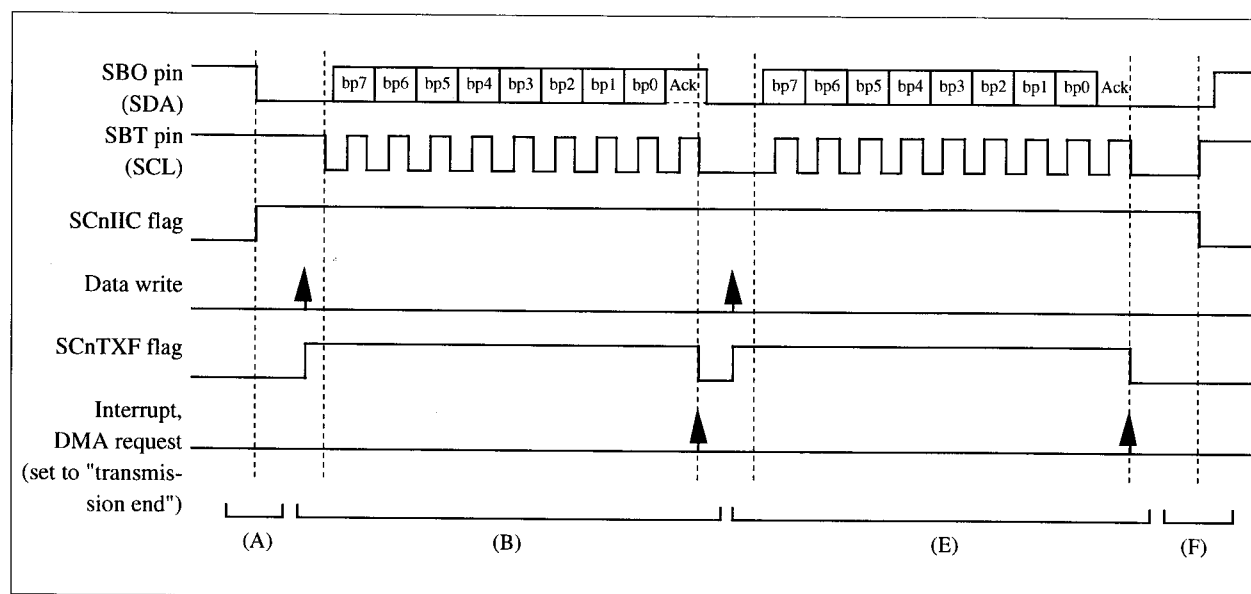
Fig. 11-5-13 Timing Chart (10)

In order to enter master reception mode, it is necessary to send the first byte in master transmission mode. Therefore, the explanation below covers the settings needed after a master transmission.

- Reception settings

  (1) Enable the reception operation. (SCnRXE)

  (2) The parity bit is used as a substitute for Ack. When receiving, set the parity bit to "'0' fixed". (SCnPB2 to 0)

(E) Data reception

  (1) Once the dummy data x' FF is written to the serial transmission buffer (SCnTXB), the clock is output and the data is received.

  The reception interrupt can be replaced with the transmission end interrupt.

  (2) After the end of reception, the SBOn pin output and the SBTn pin output are both maintained low. To continue receiving, write the dummy data x' FF to the serial transmission buffer (SCnTXB) again.

(F) Stop bit send

(1) When ending the data transmission, write a "0" to the I2C mode selection flag (SCnIIC). Be sure to write a "0" to this flag only when transmission possible (flag SCnTXF is "0") and reception buffer is empty (flag SCnRBF is "0").

(2) Simultaneously with the write, the SBTn pin output goes high. After one cycle, the SBOn pin output goes high, and the stop sequence is sent. At this point, the I2C stop sequence detection bit (SCnSPF) is set to "1".

(3) After the stop sequence is sent, disable the reception operation and initialize reception.

The I2C start sequence detection bit (SCnSTF) and the I2C stop sequence detection bit (SCnSPF) are cleared when the serial transmission buffer (SCnTXB) is written or the serial reception buffer (SCnRXB) is read.

11.5.1.6 Reception Errors

• Transfer in start-stop synchronous mode with 8-bit data length, parity on, and 1 stop bit
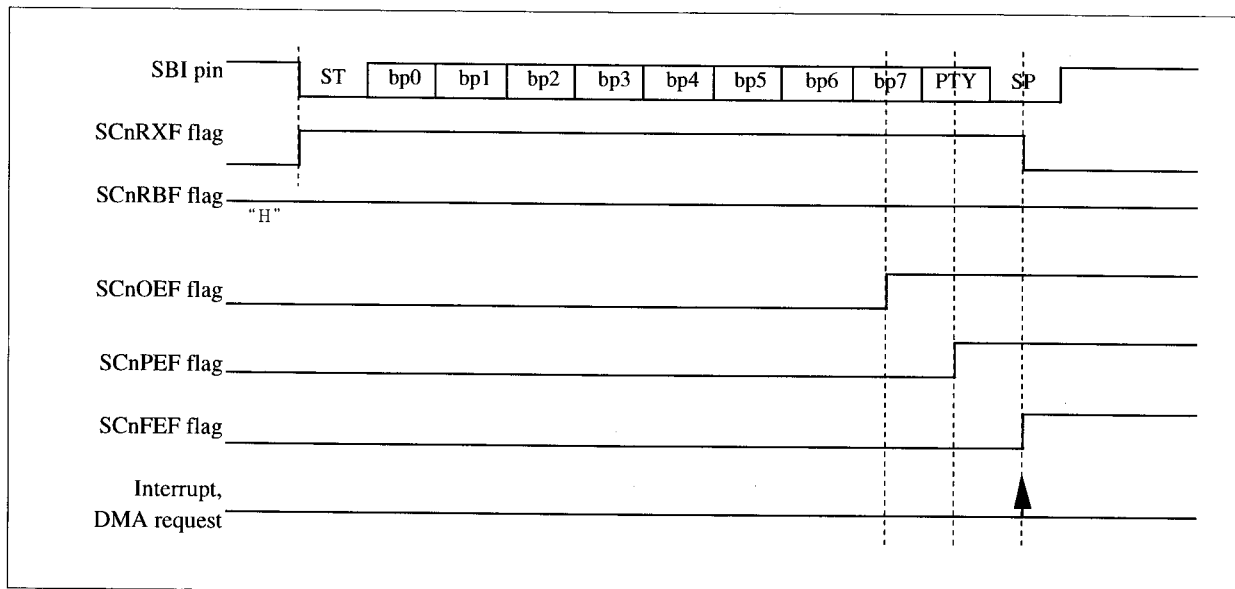
Fig. 11-5-14 Timing Chart (11)

• Transfer in clock synchronous mode with 8-bit data length and parity on
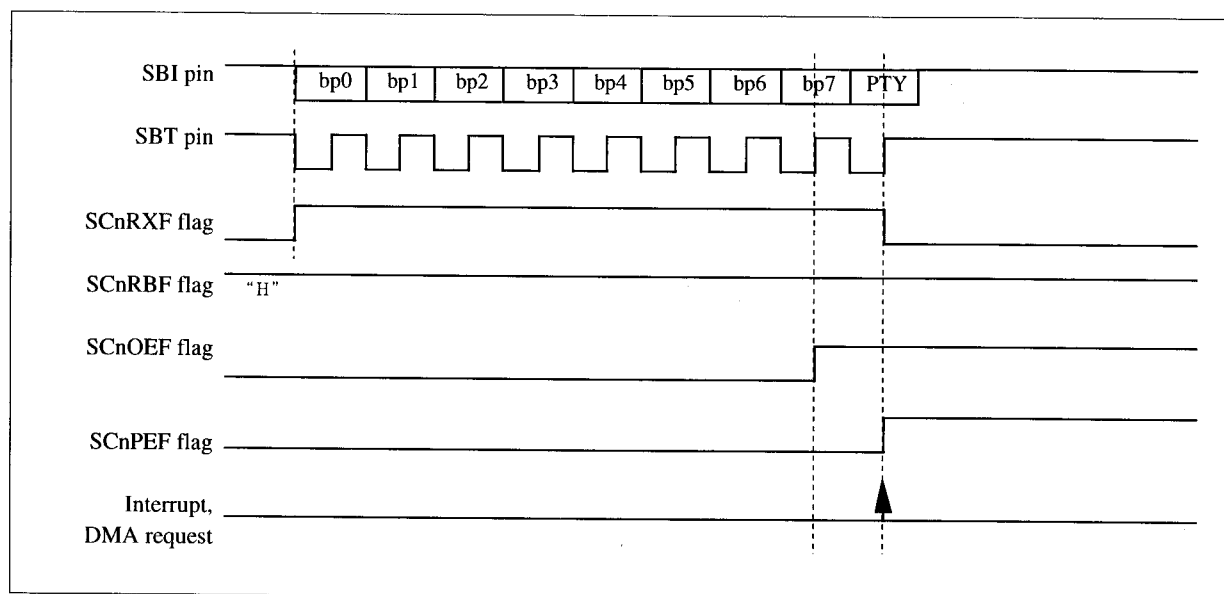
Fig. 11-5-15 Timing Chart (12)

A DMA request is generated at the moment that reception ends, regardless of whether or not an error occurred.

When "reception end" is set as the reception interrupt source, an interrupt request is generated when reception ends, regardless of whether or not an error occurred.

When "reception end with error" is set as the reception interrupt source, an interrupt request is generated when reception ends with an error having occurred. (The interrupt request is not generated at the moment that the error occurred.)

11.5.2 Serial Interface 2

11.5.2.1 Connection

<Start-stop synchronous mode>

Two different connection methods are possible, one is unidirectional transfer, and the other is bidirectional transfer.

The SBO2 pin is always an output, and the SBI2 pin is always an input.
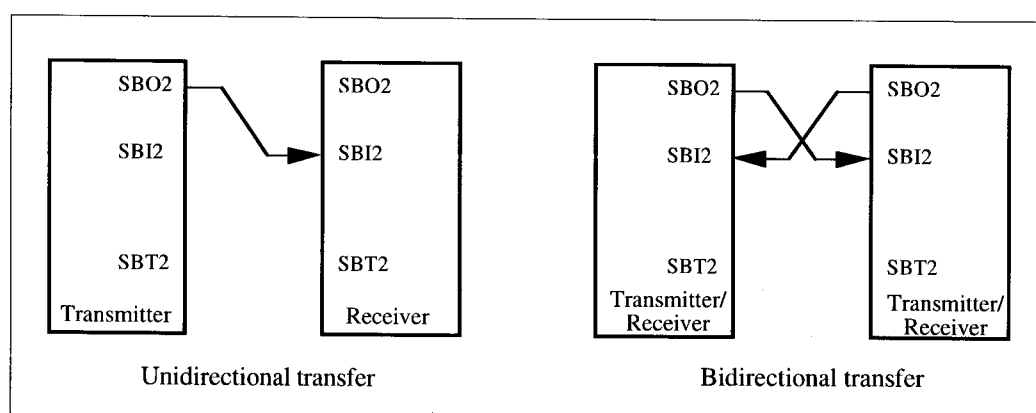
Fig. 11-5-16 Serial Interface 2 Connection Diagram

11.5.2.2 Baud Rate

Serial interface 2 is equipped with an internal 7-bit dedicated counter, and has a configuration that allows it to maintain a fast transfer speed even with a comparatively slow clock source.

For example, the following settings are recommended when using IOCLK and performing a transfer at a certain speed.

Division ratio 1 = INT (IOCLK frequency/desired baud rate/127) + 1

Division ratio 2 = INT (IOCLK frequency/desired baud rate/division ratio 1 + 0.5)

Write to SC2TIM the result that is obtained by subtracting 1 from division ratio 2. If the value of division ratio 1 is "2" or higher, timer 2 or timer 3 must be used for division. For example, set bits 0 and 1 of SC2CTR to "01", and set the timer 2 control register so that the clock is divided by the value of division ratio 1. If the value of division ratio 1 is "1," set bits 0 and 1 of SC2CTR to "00" and select IOCLK.

The baud rate that can actually be set is determined as follows:

Baud rate = IOCLK frequency/division ratio 1/division ratio 2

In this case, the baud rate error can be calculated as follows:

Baud rate error = ABS (IOCLK frequency/division ratio 1/division ratio 2/desired baud rate - 1)

Typical examples are listed in the following table.

Table 11-5-4 Baud Rate Error (1) (When IOCLK = 15 MHz)

Baud rate	High-speed UART		
	Division ratio 1	Division ratio 2	Error of transmission rate
230 400	1	65	0.16 %
115 200	2	65	0.16 %
56 000	3	89	0.32 %
38 400	4	98	0.35 %
19 200	7	112	0.35 %
9 600	13	120	0.16 %
4 800	25	125	0.00 %
2 400	50	125	0.00 %
1 200	99	126	0.21 %
600	197	127	0.08 %
300	394	127	0.08 %
150	788	127	0.08 %

Table 11-5-5 Baud Rate Error (2) (When IOCLK = 12 MHz)

Baud rate	High-speed UART		
	Division ratio 1	Division ratio 2	Error of transmission rate
230 400	1	52	0.16 %
115 200	1	104	0.16 %
56 000	2	107	0.13 %
38 400	3	104	0.16 %
19 200	5	125	0.00 %
9 600	10	125	0.00 %
4 800	20	125	0.00 %
2 400	40	125	0.00 %
1 200	79	127	0.33 %
600	158	127	0.33 %
300	315	127	0.01 %
150	630	127	0.01 %

Table 11-5-6 Baud Rate Error (3) (When IOCLK = 8 MHz)

Baud rate	High-speed UART		
	Division ratio 1	Division ratio 2	Error of transmission rate
230 400	1	34	2.12 %
115 200	1	69	0.64 %
56 000	2	71	0.60 %
38 400	2	104	0.16 %
19 200	4	104	0.16 %
9 600	7	119	0.04 %
4 800	14	119	0.04 %
2 400	27	123	0.37 %
1 200	53	126	0.17 %
600	105	127	0.01 %
300	210	127	0.01 %
150	420	127	0.01 %

Because the transfer rate error is large when transferring at a baud rate of 230.4 kbps while IOCLK = 8 MHz, using that baud rate with that IOCLK frequency is not recommended.

11.5.2.3 Transmission Interruption Function

Bits 2 and 8 of SC2CTR can be used to interrupt and resume a transmission through the status of the CTS pin.
A transmission is interrupted by the end of the transmission, by a transmission buffer empty interrupt, or by masking the DMA request signal.
Note that when controlling the transmission through the CTS pin, writing data to the transfer buffer while the interrupt signal or DMA request signal is masked is prohibited.

11.5.2.4 Notes on Use

1. When using serial interface 2, set SC2CTR before setting the other registers, and do not change the SC2CTR setting while transmission/reception is in progress or while there is data in the transmission buffer. Operation is not guaranteed if the settings are changed under such circumstances.

2. Before writing to SC2TXB, confirm that the transmission buffer is empty by checking bit 5 of SC2STR, or else set bit 4 of SC2ICR to "1" and then write to SC2TXB during the interrupt processing.

3. Set a value of 16 or greater in SC2TIM.

4. When using an external clock via the SBT2 pin, use an input clock that has a high and low pulse width that is at least twice as long as the IOCLK cycle.

5. The transmission interruption function through the CTS pin does not disable the actual transmission operation. Therefore, if there is any data remaining in the transmission buffer or the transmission shift register when a transmission was interrupted through the CTS pin, the transmission continues until all of that remaining data has been transferred, and then the transmission is interrupted.

Chapter 12. I/O Ports

12

12.1 Overview

This microcontroller has a total of 4 I/O ports: ports 0 to3 built in. These ports can all be accessed by programs as internal I/O memory space.

Ports 0 through 2 are 8-bit general-purpose input/output ports, and port 3 is a 2-bit general-purpose input/output port.

Each port also has other pin functions as described below. The control register within the port can be used to switch between the port function and the other function of each pin.

Furthermore, although not an I/O port, address output pins A(31) and A(28:26) are also used as chip select signals CS7 through 4, and like an I/O port, the control register within the port can be used to switch between the two functions of each pin. In this manual, for convenience sake, the dual-purpose address output/chip select signal pins will be referred to as "Port 4" in this manual, even though these pins do not constitute an I/O port.

Port 0 (P0)

    The pins of this port also serve as the data bus signals D(7:0).

Port 1 (P1)

    The pins of this port also serve as the data bus signals D(15:8).

Port 2 (P2)

    The pins of this port also serve as serial interface input/outputs SBT0 and SBT1, and timer input/outputs TM0IO, TM1IO, TM2IO, TM3IO, TM4IO, TM5IO, TM6IOA and TM6IOB.

Port 3 (P3)

    The pins of this port also serve as serial interface input/outputs SBO0 and SBO1.

Port 4 (P4)

    The pins of this port also serve as address outputs A(31) and A(28:26), and as chip select signals CS7 to 4. Although these pins do not constitute an I/O port, they are referred to as "Port 4" for the sake of convenience in this manual.

Table 12-1-1 lists the I/O port registers.

Table 12-1-1 List of Registers

Address	Name	Symbol	Number of bits	Initial value	Access size
x'36008000	Port 0 output register	P0OUT	8	x'00	8, 16
x'36008001	Port 1 output register	P1OUT	8	x'00	8
x'36008004	Port 2 output register	P2OUT	8	x'00	8, 16
x'36008005	Port 3 output register	P3OUT	8	x'00	8
x'36008020	Port 0 output mode register	P0MD	8	x'00	8, 16
x'36008021	Port 1 output mode register	P1MD	8	x'00	8
x'36008024	Port 2 output mode register	P2MD	8	x'FF	8, 16
x'36008025	Port 3 output mode register	P3MD	8	x'03	8
x'36008044	Port 2 dedicated output control register	P2SS	8	x'00	8
x'36008048	Port 4 dedicated output control register	P4SS	8	x'0F	8
x'36008060	Port 0 I/O control register	P0DIR	8	x'00	8, 16
x'36008061	Port 1 I/O control register	P1DIR	8	x'00	8
x'36008064	Port 2 I/O control register	P2DIR	8	x'00	8, 16
x'36008065	Port 3 I/O control register	P3DIR	8	x'00	8
x'36008080	Port 0 pin register	P0IN	8	x'XX	8, 16
x'36008081	Port 1 pin register	P1IN	8	x'XX	8
x'36008084	Port 2 pin register	P2IN	8	x'XX	8, 16
x'36008085	Port 3 pin register	P3IN	8	x'0X	8

12.2 Port 0

12.2.1 Block Diagram

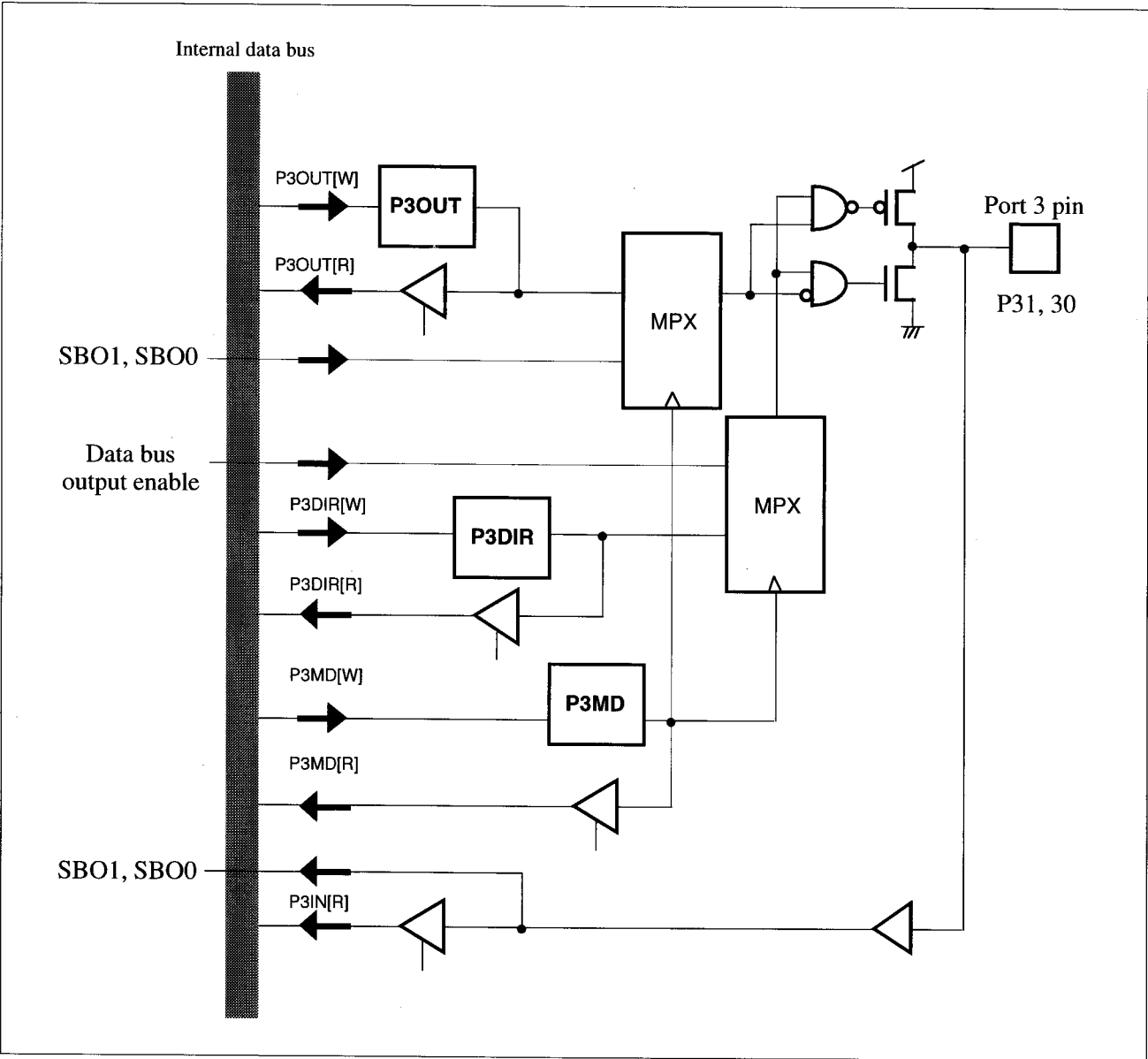Fig. 12-2-1 shows the block diagram for port 0.
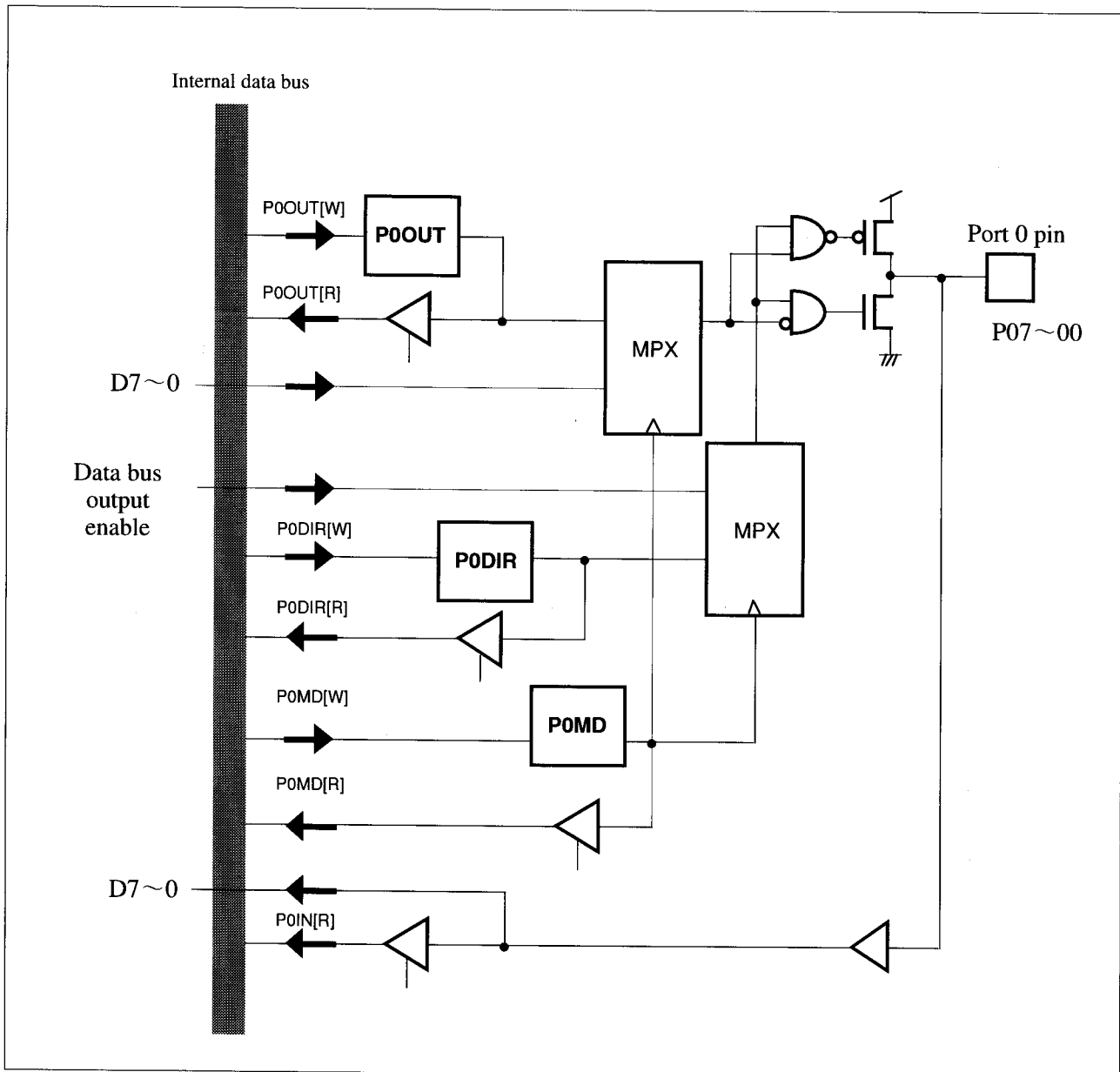
Fig. 12-2-1 Port 0 Block Diagram (P07 to 00)

12.2.2. Description of Registers

Port 0 is a general-purpose input/output port that also can be used as the data bus signals D(7:0).

Each register for port 0 is described below.

12.2.2.1 Port 0 Output Register

Register symbol: P0OUT

Address: x'36008000

Purpose: Sets the data that is to be output on port 0.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P07O	P06O	P05O	P04O	P03O	P02O	P01O	P00O
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.2.2 Port 0 Output Mode Register

Register symbol: P0MD

Address: x'36008020

Purpose: This register selects the port 0 pin output content.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P00M
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Bit No.	Bit name	Description
0	P00M	Port 0 pin output content selection 0: Data input/output (D7 to D0) 1: General-purpose I/O port (P07 to P00)
1 to 7	—	Always returns "0".

12.2.2.3 Port 0 Input/Output Control Register

Register symbol: P0DIR

Address: x'36008060

Purpose: This register sets the input/output direction of the port 0 pins.

(0: input pin; 1: output pin).

Bit No.	7	6	5	4	3	2	1	0
Bit name	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.2.4 Port 0 Pin Register

Register symbol: P0IN

Address: x'36008080

Purpose: This register is used to read the value of the port 0 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P07I	P06I	P05I	P04I	P03I	P02I	P01I	P00I
When reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

12.2.3 Pin Configuration

Table 12-2-1 shows the pin configuration of port 0.

Table 12-2-1 Port 0 Configuration

Port		P00M = 1		P00M = 0	
	P0n	P0nD = 1	P0nD = 0		
Port 0	P00	General-purpose output port	General-purpose input port	D0	Data input/output
	P01	General-purpose output port	General-purpose input port	D1	Data input/output
	P02	General-purpose output port	General-purpose input port	D2	Data input/output
	P03	General-purpose output port	General-purpose input port	D3	Data input/output
	P04	General-purpose output port	General-purpose input port	D4	Data input/output
	P05	General-purpose output port	General-purpose input port	D5	Data input/output
	P06	General-purpose output port	General-purpose input port	D6	Data input/output
	P07	General-purpose output port	General-purpose input port	D7	Data input/output

: After a reset

12.3 Port 1

12.3.1 Block Diagram

Fig. 12-3-1 shows the block diagram for port 1.
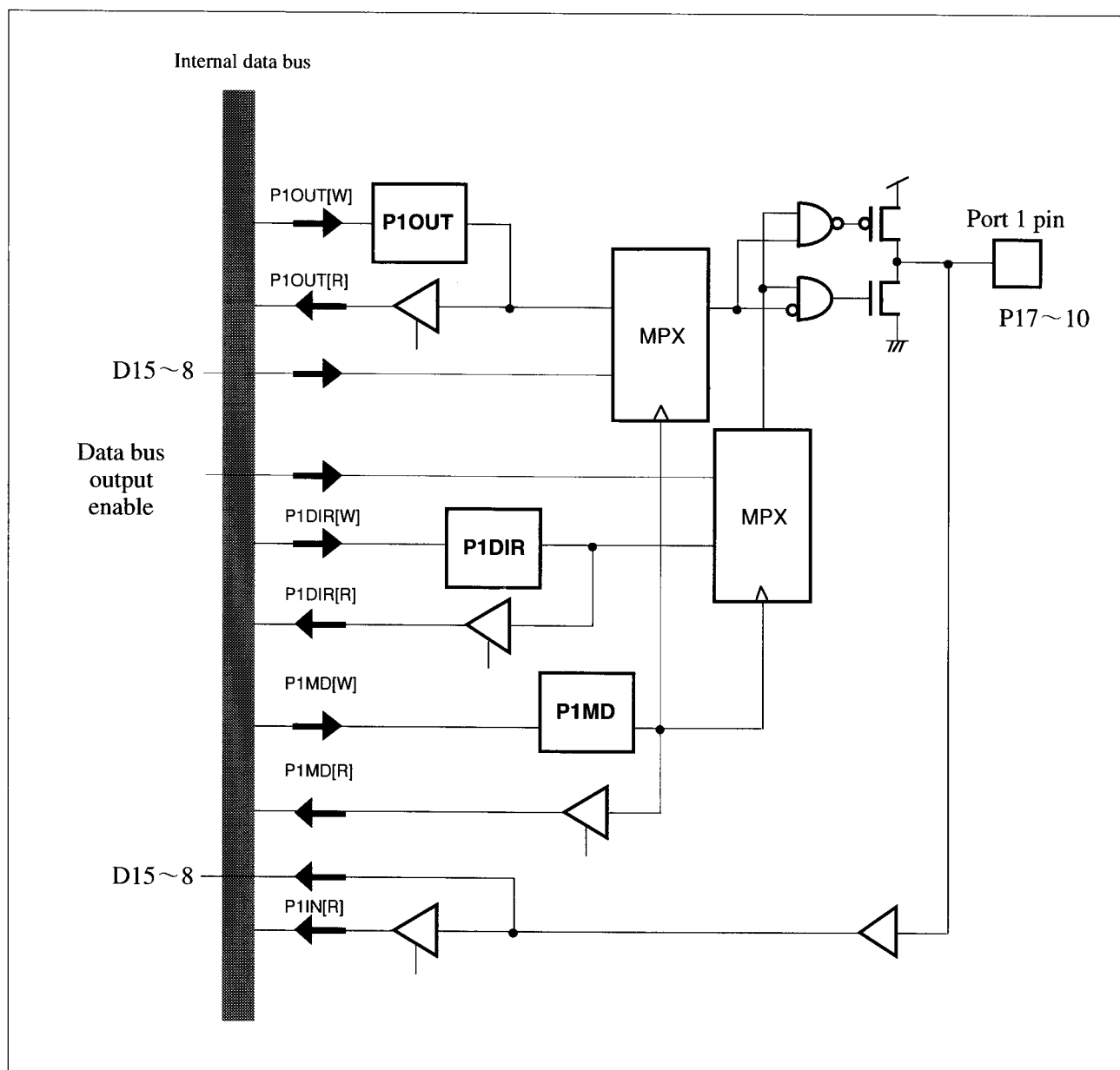
Fig. 12-3-1 Port 1 Block Diagram

12.3.2. Description of Registers

Port 1 is a general-purpose input/output port that also can be used as the data bus signals D(15:8).

Each register for port 1 is described below.

12.3.2.1 Port 1 Output Register

Register symbol: P1OUT
Address: x'36008001
Purpose: Sets the data that is to be output on port 1.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17O	P16O	P15O	P14O	P13O	P12O	P11O	P10O
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.2.2 Port 1 Output Mode Register

Register symbol: P1MD
Address: x'36008021
Purpose: This register selects the port 1 pin output content.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	-	P10M
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Bit No.	Bit name	Description
0	P10M	Port 1 pin output content selection 0: Data input/output (D15 to D8) 1: General-purpose I/O port (P17 to P10)
1 to 7	—	Always returns "0".

12.3.2.3 Port 1 Input/Output Control Register

Register symbol:	P1DIR
Address:	x'36008061
Purpose:	This register sets the input/output direction of the port 1 pins. (0: input pin; 1: output pin).

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.2.4 Port 1 Pin Register

Register symbol:	P1IN
Address:	x'36008081
Purpose:	This register is used to read the value of the port 1 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P17I	P16I	P15I	P14I	P13I	P12I	P11I	P10I
When reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

12.3.3 Pin Configuration

Table 12-3-1 shows the pin configuration of port 1.

Table 12-3-1 Port 1 Configuration

Port		P10M = 1		P10M = 0	
	P1n	P1nD = 1	P1nD = 0		
Port 1	P10	General-purpose output port	General-purpose input port	D8	Data input/output
	P11	General-purpose output port	General-purpose input port	D9	Data input/output
	P12	General-purpose output port	General-purpose input port	D10	Data input/output
	P13	General-purpose output port	General-purpose input port	D11	Data input/output
	P14	General-purpose output port	General-purpose input port	D12	Data input/output
	P15	General-purpose output port	General-purpose input port	D13	Data input/output
	P16	General-purpose output port	General-purpose input port	D14	Data input/output
	P17	General-purpose output port	General-purpose input port	D15	Data input/output

▓ : After a reset

12.4 Port 2

12.4.1 Block Diagram

Fig. 12-4-1 and Fig. 12-4-2 shows the block diagram for port 2.
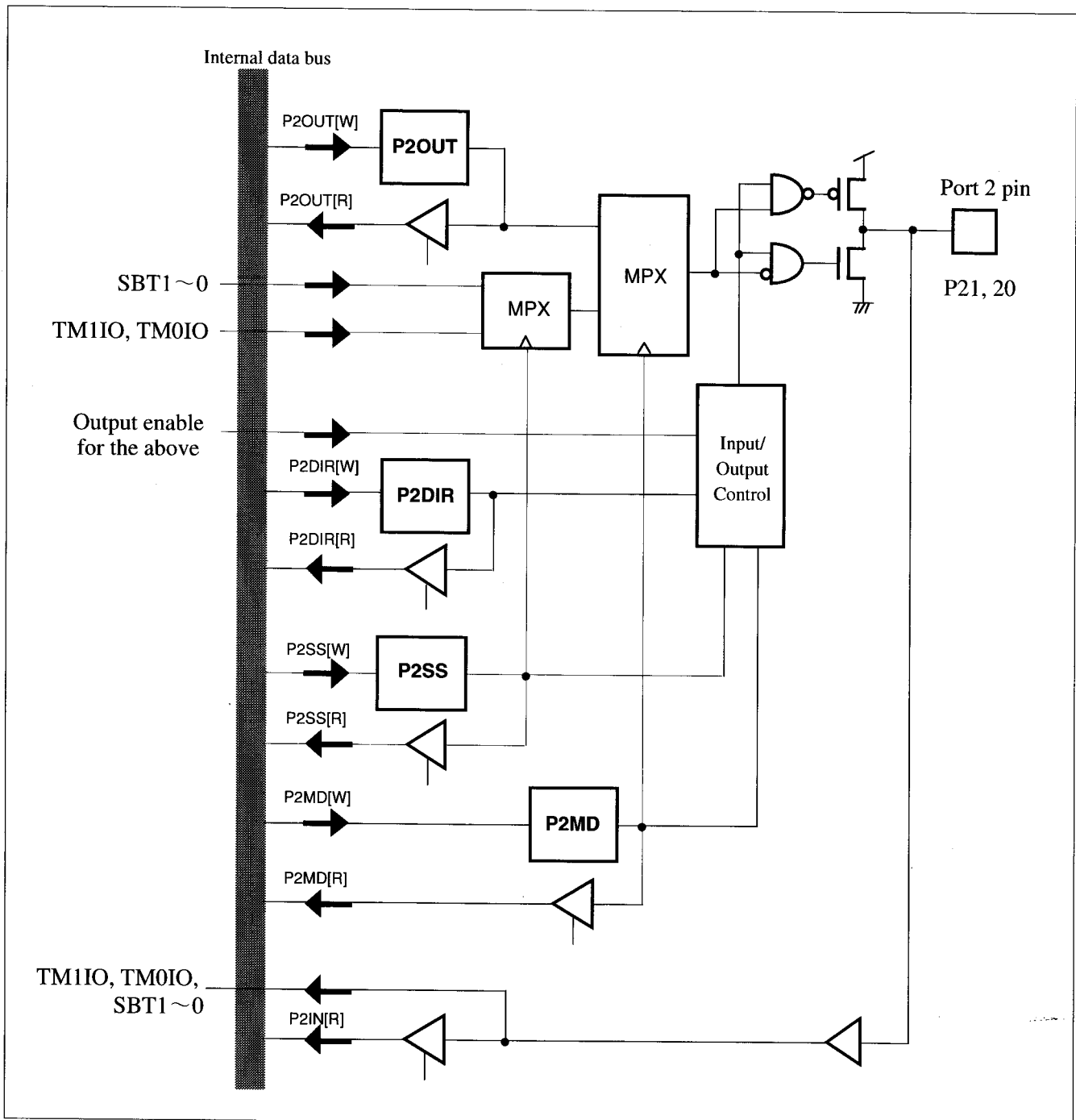
Fig. 12-4-1 Port 2 Block Diagram (P21, 20)
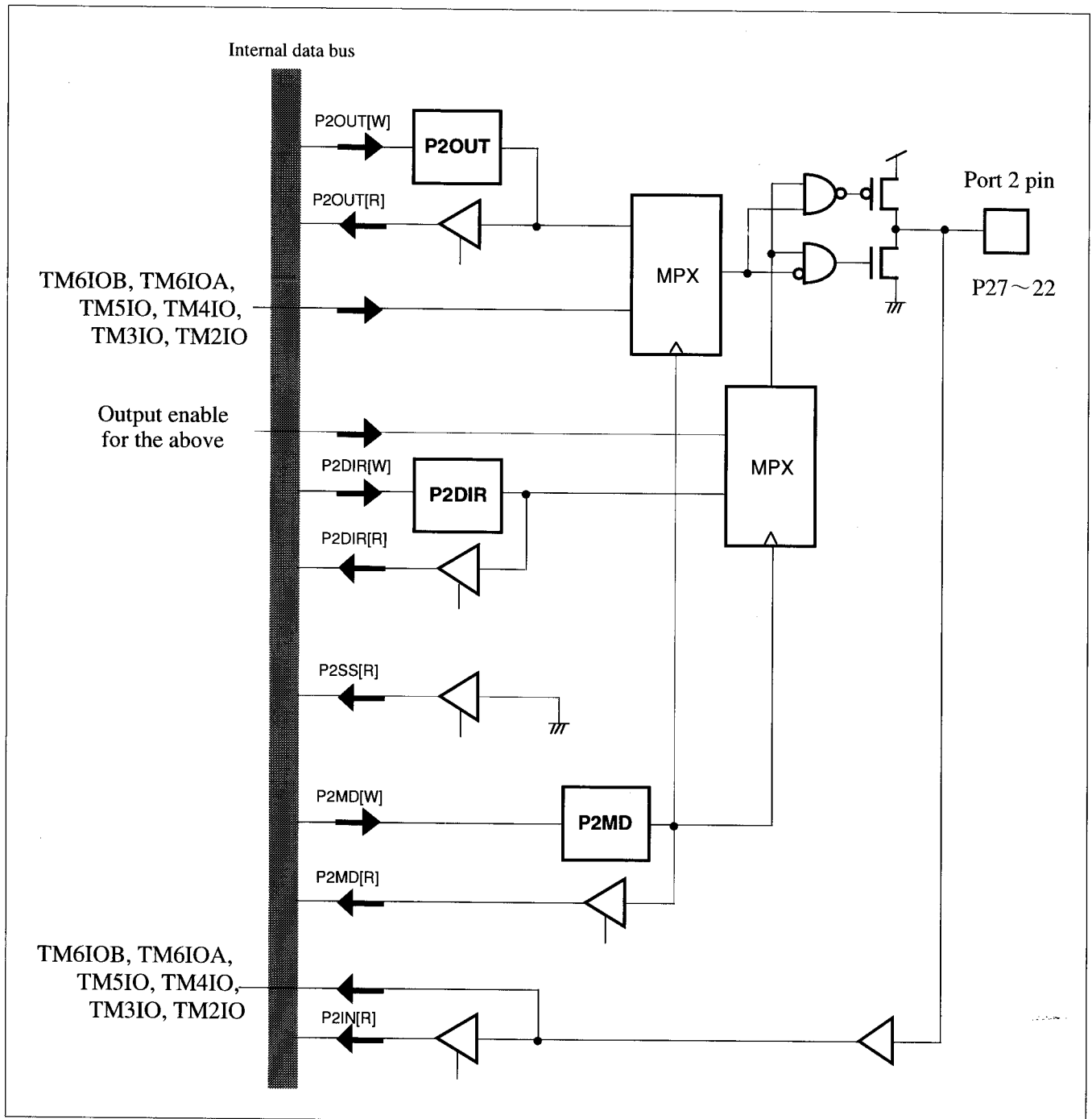
Fig. 12-4-2 Port 2 Block Diagram (P27 to 22)

12.4.2. Description of Registers

Port 2 is a general-purpose input/output port that also can be used as serial interface output SBT0 and SBT1, and timer input/outputs TM0IO, TM1IO, TM2IO, TM3IO, TM4IO, TM5IO, TM6IOA, and TM6IOB.

Each register for port 2 is described below.

12.4.2.1 Port 2 Output Register

Register symbol: P2OUT

Address: x'36008004

Purpose: Sets the data that is to be output on port 2.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27O	P26O	P25O	P24O	P23O	P22O	P21O	P20O
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.4.2.2 Port 2 Output Mode Register

Register symbol: P2MD

Address: x'36008024

Purpose: This register selects the port 2 pin output content.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27M	P26M	P25M	P24M	P23M	P22M	P21M	P20M
When reset	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details on the P27M to P20M settings, refer to the description of the port 2 dedicated output control register.

12.4.2.3 Port 2 Dedicated Output Control Register

Register symbol: P2SS

Address: x'36008044

Purpose: This register selects the port 2 pin output content in conjunction with P2MD.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P21S	P20S
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W

P27M to P22M   0: Timer input/outputs (TM2IO, TM3IO, TM4IO, TM5IO, TM6IOA, TM6IOB)

1: General-purpose I/O port (P27 to P22)

For details on other bits, see below.

P21M ; P21S   00: Serial interface input/output (SBT1)

01: Timer input/output (TM1IO)

1x: General-purpose I/O port (P21)

P20M ; P20S   00: Serial interface input/output (SBT0)

01: Timer input/output (TM0IO)

1x: General-purpose I/O port (P20)

12.4.2.4 Port 2 Input/Output Control Register

Register symbol: P2DIR

Address: x'36008064

Purpose: This register sets the input/output direction of the port 2 pins.

    (0: input pin; 1: output pin).

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D
When reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.4.2.5 Port 2 Pin Register

Register symbol: P2IN

Address: x'36008084

Purpose: This register is used to read the value of the port 2 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	P27I	P26I	P25I	P24I	P23I	P22I	P121I	P20I
When reset	X	X	X	X	X	X	X	X
Access	R	R	R	R	R	R	R	R

12.4.3   Pin Configuration

Table 12-4-1 shows the pin configuration of port 2.

Table 12-4-1 Port 2 Configuration

Port		P2nM = 1		P2nM = 0							
				P2nS = 1				P2nS = 0			
	P2n	P2nD = 1	P2nD = 0	P2nD = 1		P2nD = 0		P2nD = 1		P2nD = 0	
Port 2	P20	General-purpose output port	General-purpose input port	TM0IO	Timer 0 output	TM0IO	Timer 0 input	SBT0	Serial 0 transfer clock input/output *P2nD invalid		
	P21	General-purpose output port	General-purpose input port	TM1IO	Timer 1 output	TM1IO	Timer 1 input	SBT1	Serial 1 transfer clock input/output *P2nD invalid		
	P22	General-purpose output port	General-purpose input port					TM2IO	Timer 2 output	TM2IO	Timer 2 input
	P23	General-purpose output port	General-purpose input port					TM3IO	Timer 3 output	TM3IO	Timer 3 input
	P24	General-purpose output port	General-purpose input port					TM4IO	Timer 4 output	TM4IO	Timer 4 input
	P25	General-purpose output port	General-purpose input port					TM5IO	Timer 5 output	TM5IO	Timer 5 input
	P26	General-purpose output port	General-purpose input port					TM6IOA	Timer 6 output	TM6IOA	Timer 6 input
	P27	General-purpose output port	General-purpose input port					TM6IOB	Timer 6 output	TM6IOB	Timer 6 input

▨ : After a reset

12.5 Port 3

12.5.1 Block Diagram

Fig. 12-5-1 shows the block diagram for port 3.

Fig. 12-5-1 Port 3 Block Diagram (P31, 30)

12.5.2. Description of Registers

Port 3 is a general-purpose input/output port that also can be used as the serial data input/outputs SBO0 and SBO1.

Each register for port 3 is described below.

12.5.2.1 Port 3 Output Register

Register symbol: P3OUT

Address: x'36008005

Purpose: Sets the data that is to be output on port 3.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P31O	P30O
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W

12.5.2.2 Port 3 Output Mode Register

Register symbol: P3MD

Address: x'36008025

Purpose: This register selects the port 3 pin output content.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P31M	P30M
When reset	0	0	0	0	0	0	1	1
Access	R	R	R	R	R	R	R/W	R/W

Bit No.	Bit name	Description
0	P30M	Port 3 pin output content selection
		0: Serial data input/output (SBO0)
		1: General-purpose I/O port (P30)
1	P31M	Port 3 pin output content selection
		0: Serial data input/output (SBO1)
		1: General-purpose I/O port (P31)
2 to 7	—	Always returns "0".

12.5.2.3 Port 3 Input/Output Control Register

Register symbol: P3DIR

Address: x'36008065

Purpose: This register sets the input/output direction of the port 3 pins.

(0: input pin; 1: output pin).

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P31D	P30D
When reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W

12.5.2.4 Port 3 Pin Register

Register symbol: P3IN

Address: x'36008085

Purpose: This register is used to read the value of the port 3 pins.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	-	-	P31I	P30I
When reset	0	0	0	0	0	0	X	X
Access	R	R	R	R	R	R	R	R

12.5.3 Pin Configuration

Table 12-5-1 shows the pin configuration of port 3.

Table 12-5-1 Port 3 Configuration

Port		P3nM = 1		P3nM = 0	
	P3n	P3nD = 1	P3nD = 0		
Port 3	P30	General-purpose output port	General-purpose input port	SBO0	Serial 0 data input/output *P3nD invalid
	P31	General-purpose output port	General-purpose input port	SBO1	Serial 1 data input/output *P3nD invalid

▓▓▓ : After a reset

12.6 Port 4

12.6.1 Description of Resisters

Port 4 consists of dual-purpose pins that are used for address outputs A(31) and A(28:26), and chip select signals CS7 to 4. Although these pins do not actually form an I/O port, for the sake of convenience they are referred to as "port 4" in this manual. Like an I/O port, the control register within the port can be used to switch between the two functions of each pin.

The register that is used for switching the function of each pin is described below.

12.6.1.1 Port 4 Dedicated Output Control Register

Register symbol: P4SS
Address: x'36008048
Purpose: This port selects the pin output content.

Bit No.	7	6	5	4	3	2	1	0
Bit name	-	-	-	-	P43S	P42S	P41S	P40S
When reset	0	0	0	0	1	1	1	1
Access	R	R	R	R	R/W	R/W	R/W	R/W

Bit No.	Bit name	Description
0	P40S	Pin output content selection
		0: Address output (A31)
		1: Chip select signal (CS7)
1	P41S	Pin output content selection
		0: Address output (A28)
		1: Chip select signal (CS6)
2	P42S	Pin output content selection
		0: Address output (A27)
		1: Chip select signal (CS5)
3	P43S	Pin output content selection
		0: Address output (A26)
		1: Chip select signal (CS4)
4 to 7	—	Always returns "0".

12.6.2 Pin Configuration

Table 12-6-1 shows the pin configuration for port 4.

Table 12-6-1 Port 4 Configuration

Port	P4n	P4nM = 1		P4nM = 0	
Port 4	P40	CS7	Chip select signal	A31	Address output
	P41	CS6	Chip select signal	A28	Address output
	P42	CS5	Chip select signal	A27	Address output
	P43	CS4	Chip select signal	A26	Address output

▨ : After a reset

Chapter 13.   Electrical Characteristics

> **!** This LSI manual describes the standard specifications.
> When using this LSI, contact our sales department for the product standards.

13.1 Absolute Maximum Ratings

Table 13-1-1 shows the absolute maximum ratings.

Table 13-1-1 Absolute Maximum Ratings

Vss, PVss = 0.0 V

	Item	Symbol	Ratings			Unit
A1	Supply voltage	V_{DD}	- 0.3	to	+ 4.6	V
A2	PLL supply pin (PVDD) voltage	PV_{DD}	- 0.3	to	+ 4.6	V
A3	5 V supply pin (VREF5) voltage[*1]	V_{DD5}	- 0.3	to	+ 5.7	V
A4	Input pin voltage 1	V_I	- 0.3	to	V_{DD} + 0.3	V
A5	Input pin voltage 2	V_{I5}	- 0.3	to	+ 6.0	V
A6	Output pin voltage	V_O	- 0.3	to	V_{DD} + 0.3	V
A7	Input/output pin voltage	V_{IO}	- 0.3	to	V_{DD} + 0.3	V
A8	Operating ambient temperature	T_A	- 20	to	+ 70	°C
A9	Storage temperature	T_{STR}	- 55	to	+ 125	°C
A10	Allowable power dissipation	P_D	0.9			W
A11	Averaged output current	IO	± 12			mA

Notes:

(1) The absolute maximum ratings are the allowable values which, if applied to the chip, will not cause damage to the chip; they are not the values at which operation is guaranteed.

(2) Connect all VDD and PVDD pins directly to an external power supply.

(3) Connect all VSS and PVSS pins directly to an external ground.

(4) Insert at least one bypass capacitor of at least 0.33 µF between the power supply pins and ground.

[*1] The rising edge and falling edge sequences must observe the criteria described below.

t_{REF5S} and t_{REF5H} must be at least 0.
Transitions in VDD and VREF5 must be smooth.
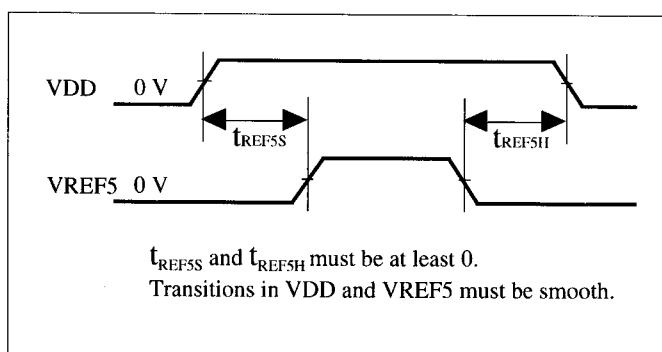
13.2 Operating Conditions

Table 13-2-1 shows the operating conditions.

Table 13-2-1 Operating Conditions

$V_{SS}, PV_{SS} = 0.0$ V

$T_A = -20$ ℃ to + 70 ℃

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
B1	Supply voltage	V_{DD}	-	3.135	3.3	3.465	V
B2	PLL supply voltage	PV_{DD}	-	3.135	3.3	3.465	V
B3	5 V supply voltage	V_{DD5}	-	V_{DD}	5.0	5.25	V
Crystal oscillation (OSCI)							
B4	Oscillating frequency	F_{OSC}	FRQS pin = H level	13.0	-	16.6	MHz
			FRQS pin = L level	26.0	-	33.3	

Note:

(5) VDD and PVDD must always be the same voltage.

13.3 DC Characteristics

Tables 13-3-1 through 13-3-4 show the DC characteristics.

Table 13-3-1 DC Characteristics (1)

Vss, PVss = 0.0 V

$T_A = -20\ °C\ to + 70\ °C$

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
C1	Supply current during operation	I_{DD1}	V_{DD}, PV_{DD} = 3.3 V V_I = V_{DD} or V_{SS} F_{OSC} = 16.7 MHz FRQS pin = H level Output open	-	-	250	mA
C2	Supply current in SLEEP mode (VDD pin)	I_{DD2}	V_{DD}, PV_{DD} = 3.465 V V_I = V_{DD} or V_{SS} F_{OSC} = 16.7 MHz FRQS pin = H level Output open	-	-	50	mA
C3	Supply current in HALT mode (VDD pin)	I_{DD3}	V_{DD}, PV_{DD} = 3.465 V V_I = V_{DD} or V_{SS} F_{OSC} = 16.7 MHz FRQS pin = H level Output open	-	-	6	mA
C4	Supply current when stopped (VDD pin)	I_{DD4}	V_{DD}, PV_{DD} = 3.465 V V_I = V_{DD} or V_{SS} F_{OSC} = Oscillation stopped Output open	-	-	1.25	mA
C5	Supply current during PLL operation (PVDD pin)	I_{PDD1}	V_{DD}, PV_{DD} = 3.3 V V_I = V_{DD} or V_{SS} F_{OSC} = 16.7 MHz FRQS pin = H level	-	-	3.0	mA
C6	Supply current when PLL is stopped (PVDD pin)	I_{PDD2}	V_{DD}, PV_{DD} = 3.465 V V_I = V_{DD} or V_{SS} F_{OSC} = Oscillation stopped	-	-	10.0	µA
C7	Supply Current of 5 V supply pin (VREF5 pin)	I_{VREF5}	V_{DD}, PV_{DD} = 3.465 V VREF5 = 5.25 V V_I = V_{DD} or V_{SS} F_{OSC} = Oscillation stopped	-	-	10.0	µA

Table 13-3-2 DC Characteristics (2)

$$V_{DD}, PV_{DD} = 3.3\ V \pm 0.165\ V$$

$$V_{DD5} = 5\ V$$

$$V_{SS}, PV_{SS} = 0.0\ V$$

$$T_A = -20\ °C\ to + 70\ °C$$

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
Input/output pins <Output: Push-pull/Input: TTL 5 V level> A0 to A10, RD, SIZE0 to SIZE1, D0 to 31, DSCLK, DSDAT, SBO0 to SBO1, TM0IO to TM5IO, TM6IOA, TM6IOB							
C7	Input voltage high level	V_{IH1}	-	2.2	-	V_{DD5}	V
C8	Input voltage low level	V_{IL1}	-	0.0	-	0.6	V
C9	Output voltage high level	V_{OH1}	$V_I = V_{DD}$ or V_{SS} $I_O = -4\ mA$	$V_{DD} - 0.6$	-	-	V
C10	Output voltage low level	V_{OL1}	$V_I = V_{DD}$ or V_{SS} $I_O = 4\ mA$	-	-	0.4	V
C11	Output leak current	I_{OZ1}	$V_O = Hi\text{-}Z$ state	- 10	-	10	μA

Table 13-3-3 DC Characteristics (3)

V_{DD}, PV_{DD} = 3.3 V ± 0.165 V

V_{DD5} = 5 V

V_{SS}, PV_{SS} = 0.0 V

T_A = -20 °C to + 70 °C

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
Output pins <Output: Push-pull> A11 to A31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{CAS0}$ to $\overline{CAS3}$, $\overline{WE0}$ to $\overline{WE3}$, $\overline{AMC}$, $\overline{RE}$, $\overline{BCK}$, $\overline{BG}$, WDOVF, SYSCLK, $\overline{DMK0}$ to $\overline{DMK3}$, SBO2							
C12	Output voltage high level	V_{OH2}	V_I = V_{DD} or V_{SS} I_O = - 4 mA	V_{DD} - 0.6	-	-	V
C13	Output voltage low level	V_{OL2}	V_I = V_{DD} or V_{SS} I_O = 4 mA	-	-	0.4	V
C14	Output leak current	I_{OZ2}	V_O = Hi-Z state	- 10	-	10	µA
Input pins <Input: TTL 5 V level> $\overline{IRQ0}$ to $\overline{IRQ7}$, $\overline{NMIRQ}$, $\overline{DK}$, $\overline{BCR}$, $\overline{BR}$, $\overline{FRQS}$, $\overline{DMR0}$ to $\overline{DMR3}$, BMODE, MMODE, $\overline{RST}$, $\overline{CTS}$, SBI0 to SBI2, SBT2							
C15	Input voltage high level	V_{IH3}	-	2.2	-	V_{DD5}	V
C16	Input voltage low level	V_{IL3}	-	0.0	-	0.6	V
C17	Input leak current	I_{OZ3}	V_I = V_{SS} to V_{DD}	- 10	-	10	µA

Table 13-3-4 DC Characteristics (4)

VDD, PVDD = 3.3 V ± 0.165 V

VSS, PVSS = 0.0 V

TA = -20 °C to + 70 °C

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
OSCI pin (when using external clock input): Refer to Fig. 13-3-1 in the case of self-excited oscillation by crystal.							
C18	Input voltage high level	V_{IH4}	-	$V_{DD} \times 0.7$	-	V_{DD}	V
C19	Input voltage low level	V_{IL4}	-	0.0	-	$V_{DD} \times 0.3$	V
Pin capacitance							
C20	Input pin	C_{IN}	$V_{DD} = V_{DD5} = V_I = 0$ V	-	7	15	pF
C21	Output pin	C_{OUT}	Ta = 25 °C	-	7	15	pF
C22	Input/output pin	$C_{I/O}$		-	7	15	pF

* Use C1 and C2 that will result
  in the optimal circuit characteristics.
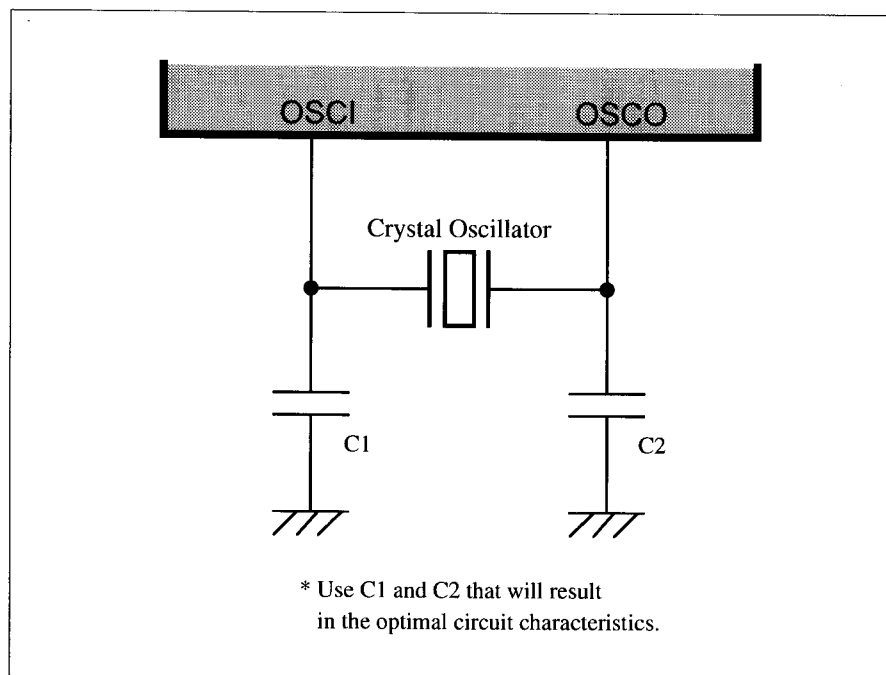
Fig. 13-3-1 Recommended Self-excited Oscillation Circuit

13.4 AC Characteristics

Tables 13-4-1 through 13-4-12 show the AC characteristics.

13.4.1 Reset Signal Timing

Table 13-4-1 AC Characteristics (1)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$

$V_{SS} = 0 \text{ V}$

$T_A = -20 \text{ °C to} + 70 \text{ °C}$

$C_L = 50 \text{ pF}$

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
Reset input timing (Refer to Figs. 13-4-1 and 13-4-2.)							
E1	Reset signal pulse width (RST)	t_{RSTW}	FRQS pin = low	5.5 t_{CYC}	-	-	ns
			FRQS pin = high	2.5 t_{CYC}	-	-	
E2	Power on oscillation stabilization time (VDD)	t_{RSTN}	-	1	-	-	ms
E3	Input clock frequency switching signal setup time (FRQS)	$t_{RSTFRQS}$	-	0.5	-	-	ms
E4	Mode setting signal setup time (MMODE)	t_{RSTMMS}	-	0.5	-	-	ms
E5	Bus mode switching signal setup time (BMODE)	t_{RSTBMS}	-	0.5	-	-	ms
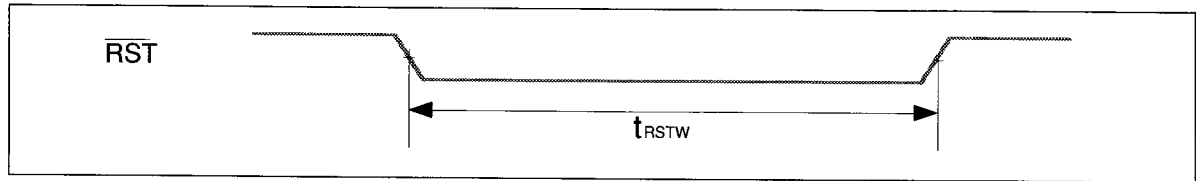
Fig. 13-4-1 Reset Timing (1)
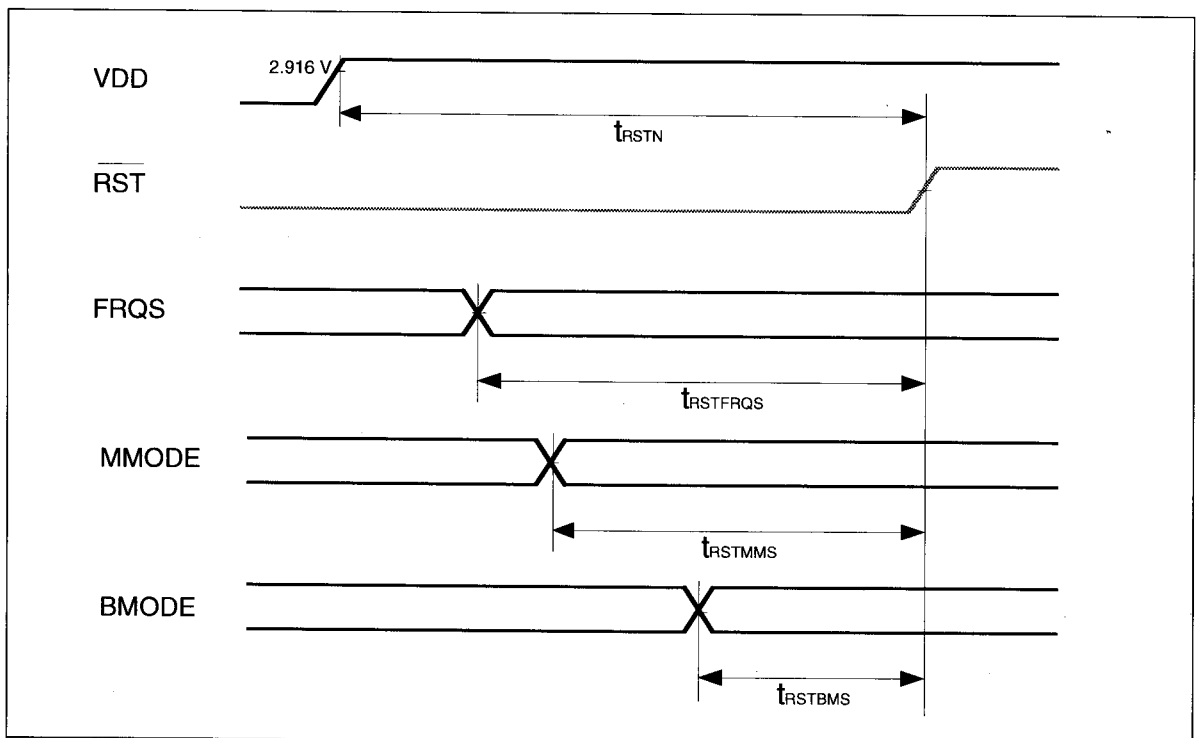
Fig. 13-4-2 Reset Timing (2)

13.4.2 Clock Timing

Table 13-4-2 AC Characteristics (2)

$$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$$
$$V_{SS} = 0 \text{ V}$$
$$T_A = -20 \text{ °C to} + 70 \text{ °C}$$
$$C_L = 50 \text{ pF}$$

Item		Symbol	Conditions	Allowable values			Unit
				Min.	Typ.	Max.	
Clock timing (Refer to Fig. 13-4-3.)							
E6	External clock input cycle time	t_{EXCCYC}	FRQS pin = L level	30.0	-	38	ns
			FRQS pin = H level	60.0	-	76	
E7	External clock input pulse width - high	t_{EXCH}	-	$\dfrac{t_{EXCCYC}}{2} - 5$	-	-	ns
E8	External clock input pulse width - low	t_{EXCL}	-	$\dfrac{t_{EXCCYC}}{2} - 5$	-	-	ns
E9	External clock input rising time	t_{EXCR}	-	-	-	5	ns
E10	External clock input falling time	t_{EXCF}	-	-	-	5	ns
E11	System clock output pulse width + transition time (SYSCLK)	t_{CYCH}	-	$\dfrac{t_{CYC}}{2} - 6$	-	$\dfrac{t_{CYC}}{2} + 6$	ns
E12	System clock output cycle time (SYSCLK)	t_{CYC}	-	$t_{EXCCYC} - 1$	t_{EXCCYC}	$t_{EXCCYC} + 1$	ns
E13	System clock output pulse width - high (SYSCLK)	t_{CH}	-	$\dfrac{t_{CYC}}{2} - 9$	-	-	ns
E14	System clock output pulse width - low (SYSCLK)	t_{CL}	-	$\dfrac{t_{CYC}}{2} - 9$	-	-	ns
E15	System clock output rising time (SYSCLK)	t_{CR}	-	-	-	9	ns
E16	System clock output falling time (SYSCLK)	t_{CF}	-	-	-	9	ns
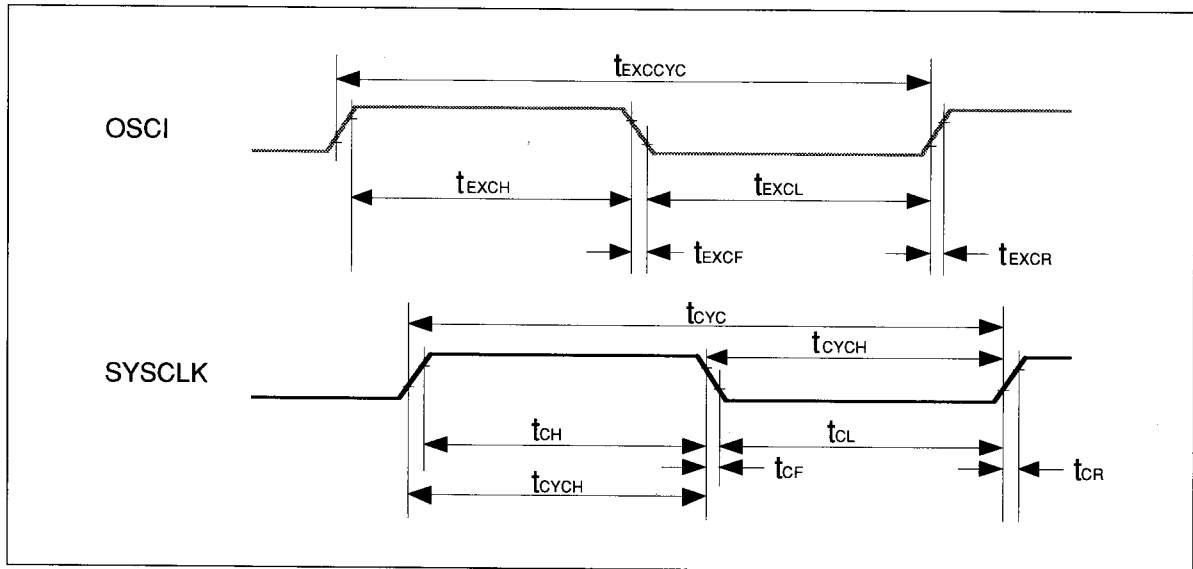
Fig. 13-4-3 System Clock Timing

13.4.3   Address/Data Transfer Signal Timing

Table 13-4-3 AC Characteristics (3)

$V_{DD} = 3.3\ V \pm 0.165\ V$

$V_{SS} = 0\ V$

$T_A = -20\ °C\ to\ +70\ °C$

$C_L = 50\ pF$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Synchronous mode data transfer signal output timing (Refer to Fig. 13-4-4.)						
E17	Address delay time (A31 - 0, RD/$\overline{WT}$, SIZE1 - 0)	t_{AD}	5	$\dfrac{t_{CYC}}{4} \times 4$	5	ns
E18	Address hold time 1 (A31 - 0, RD/$\overline{WT}$, SIZE1 - 0)	t_{AH1}	-	- 5	10	ns
E19	Address hold time 2 (A31 - 0, RD/$\overline{WT}$, SIZE1 - 0)	t_{AH2}	-	$\dfrac{t_{CYC}}{2} - 5$	-	ns
E20	Address hold time 3 (A31 - 0, RD/$\overline{WT}$, SIZE1 - 0)	t_{AH3}	-	$\dfrac{t_{CYC}}{2} - 5$	-	ns
E21	Chip select signal fall delay time ($\overline{CS7 - 0}$)	t_{CSDF}	-	- 5	2	ns
E22	Chip select signal rise delay time ($\overline{CS7 - 0}$)	t_{CSDR}	-	- 5	2	ns
E23	Chip select signal hold delay time 1 ($\overline{CS7 - 0}$)	t_{CSH1}	-	$\dfrac{t_{CYC}}{2} - 5$	-	ns
E24	Chip select signal hold delay time 2 ($\overline{CS7 - 0}$)	t_{CSH2}	-	$\dfrac{t_{CYC}}{2} - 5$	-	ns
E25	Read data enable signal fall delay time ($\overline{RE}$)	t_{REDF}	-	- 5	5	ns
E26	Read data enable signal rise delay time ($\overline{RE}$)	t_{REDR}	-	- 5	5	ns
E27	Write data enable signal fall delay time ($\overline{WE3 - 0}$)	t_{WEDF}	-	- 5	5	ns
E28	Write data enable signal rise delay time ($\overline{WE3 - 0}$)	t_{WEDR}	-	- 5	5	ns
E29	Write data delay time (D31 - 0)	t_{WDD}	-	-	5	ns
E30	Write data hold delay time 1 (D31 - 0)	t_{WDH1}	-	- 5	-	ns

Table 13-4-4 AC Characteristics (4)

$V_{DD} = 3.3\ V \pm 0.165\ V$

$V_{SS} = 0\ V$

$T_A = -20\ °C\ to + 70\ °C$

$C_L = 50\ pF$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Synchronous mode data transfer signal output timing (Refer to Fig. 13-4-4.)						
E31	Write data hold time 2 (D31 - 0)	t_{WDH2}	-	$\dfrac{t_{CYC}}{2} - 5$	-	ns
E32	Read data setup time (D31 - 0)	t_{RDS}	-	12	-	ns
E33	Read data hold time (D31 - 0)	t_{RDH}	-	0	-	ns
E34	Data acknowledge signal setup time ($\overline{DK}$)	t_{DKS}	-	12	-	ns
E35	Data acknowledge signal hold time ($\overline{DK}$)	t_{DKH}	-	0	-	ns
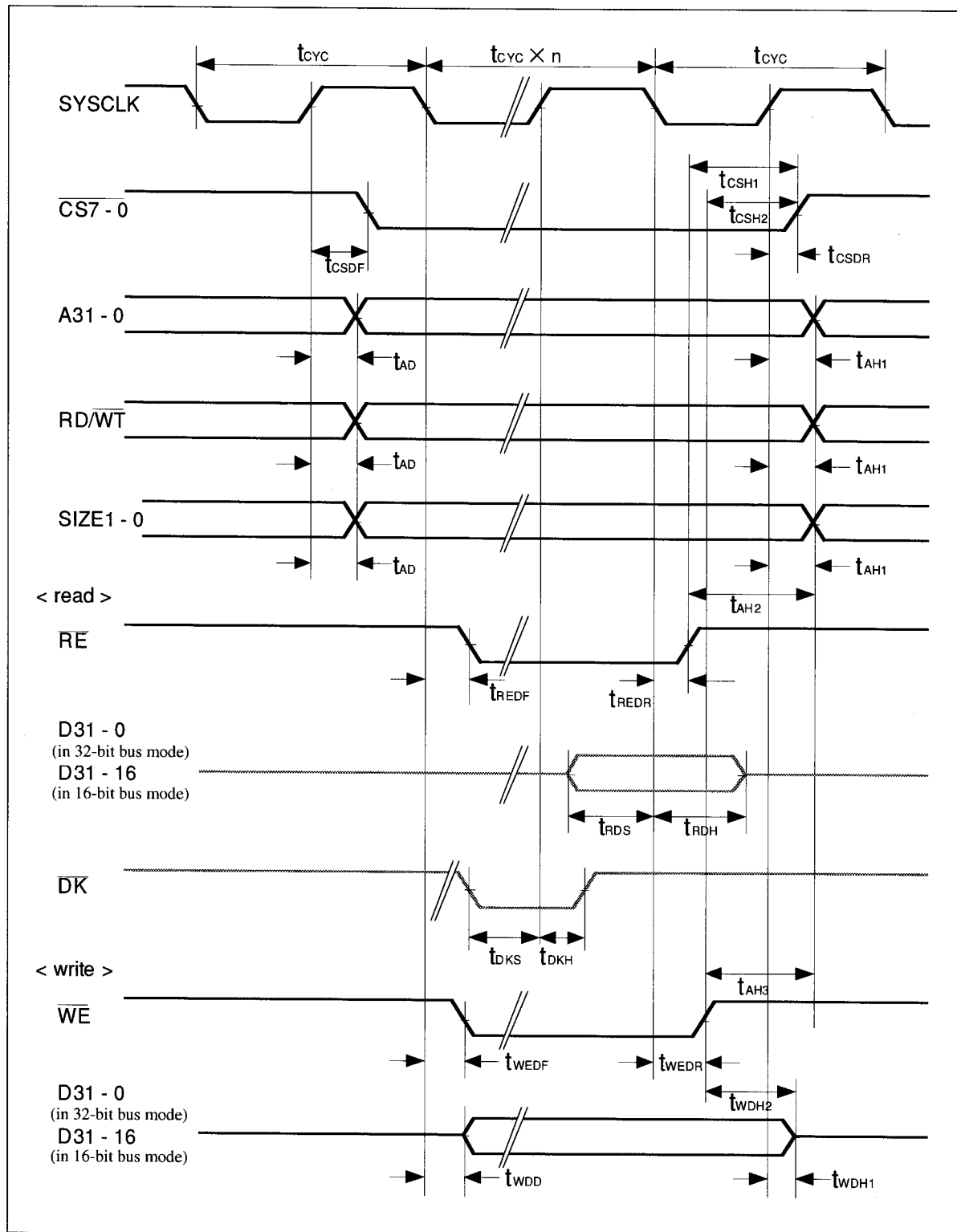
Fig. 13-4-4 Synchronous Mode Data Transfer Signal Timing
("n" represents the number of wait states.)

Table 13-4-5 AC Characteristics (5)

$$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$$
$$V_{SS} = 0 \text{ V}$$
$$T_A = -20 \text{ °C to } + 70 \text{ °C}$$
$$C_L = 50 \text{ pF}$$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Asynchronous mode data transfer signal output timing (Refer to Fig. 13-4-5.)						
E36	Chip select signal fall delay time (CS7 - 0)	t_{CSDFA}	-	- 9	0	ns
E37	Chip select signal rise delay time (CS7 - 0)	t_{CSDRA}	-	- 5	4	ns
E38	Read data enable signal fall delay time (RE)	t_{REDFA}	-	-	$\dfrac{t_{CYC}}{4} + 2$	ns
E39	Read data enable signal rise delay time (RE)	t_{REDRA}	-	-	4	ns
E40	Write data enable signal fall delay time (WE3 - 0)	t_{WEDFA}	-	-	$\dfrac{t_{CYC}}{4} + 2$	ns
E41	Write data enable signal rise delay time (WE3 - 0)	t_{WEDRA}	-	-	4	ns
E42	Write data delay time (D31 - 0)	t_{WDDA}	-	-	5	ns
E43	Write data hold delay time (D31 - 0)	t_{WDHA}	-	$\dfrac{t_{CYC}}{4} - 5$	-	ns
E44	Read data setup time (D31 - 0)	t_{RDSA}	-	18	-	ns
E45	Read data hold time (D31 - 0)	t_{RDHA}	-	0	-	ns
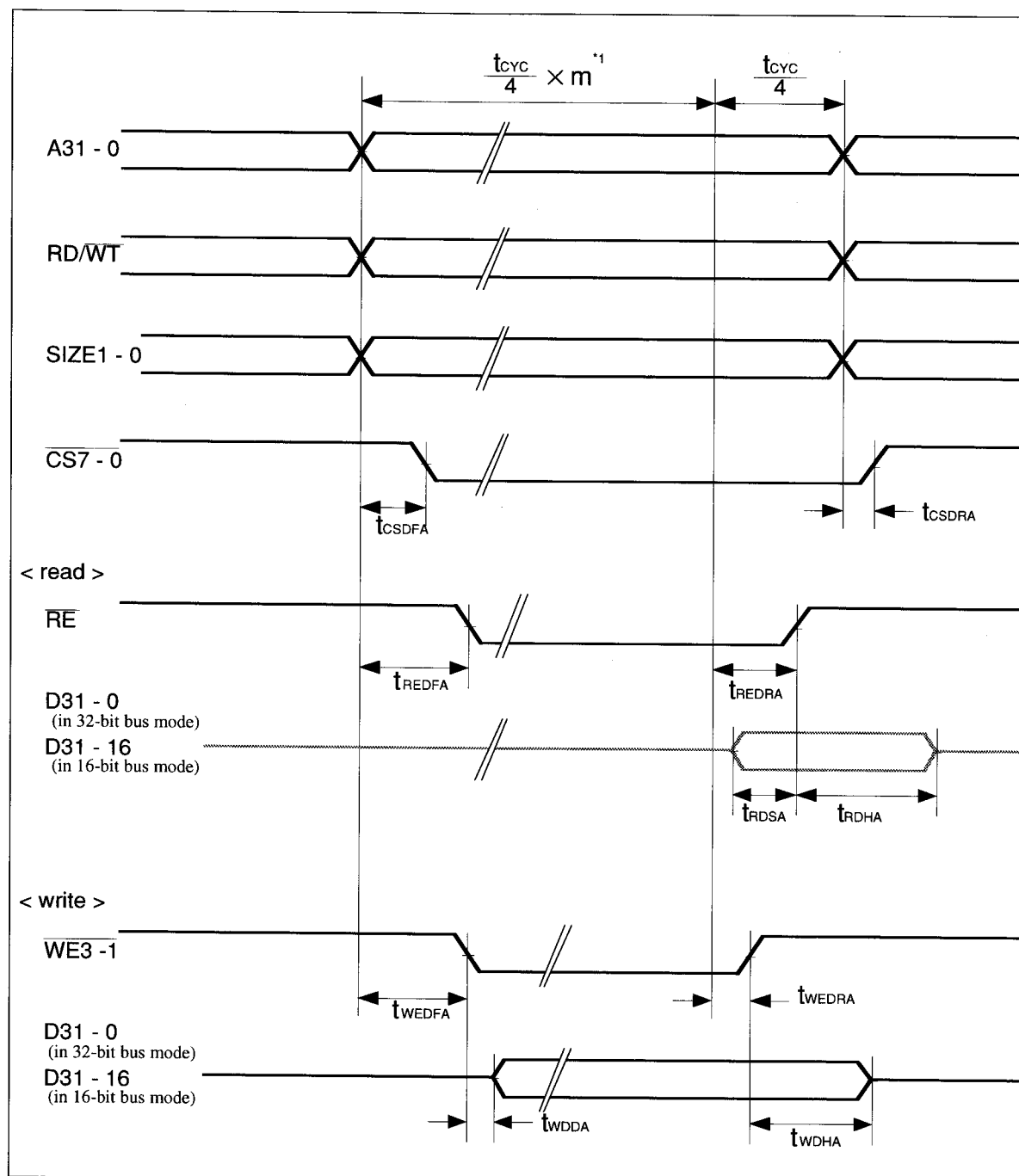
Fig. 13-4-5 Asynchronous Mode Data Trasnsfer Signal Timing

Notes:

m in *1 changes depending on the setting contents of MEMCTR0.

$$m = \begin{cases} n & (\text{FRQ} = 1) \\ 2n + 1 & (\text{FRQ} = 0) \end{cases}$$

(FRQ = 0) ("n" represents the number of wait states.)

Table 13-4-6 AC Characteristics (6)

$$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$$
$$V_{SS} = 0 \text{ V}$$
$$T_A = -20 \text{ °C to} + 70 \text{ °C}$$
$$C_L = 50 \text{ pF}$$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
DRAM mode data transfer signal output timing (Refer to Fig. 13-4-6.)						
E46	Row Address output with (A31 - 0)	t_{ROW}	-	$\dfrac{t_{CYC}}{4}$ x n_R [*1] -5	-	ns
E47	Column address output with (A31 - 0)	t_{COL}	-	$\dfrac{t_{CYC}}{4}$ x n_C [*1] -5	-	ns
E48	Row address strobe signal fall delay time ($\overline{RAS4 - 1}$)	t_{RASDF}	-	$\dfrac{t_{CYC}}{4}$ [*1] -8	$\dfrac{t_{CYC}}{4}$ [*1] $+2$	ns
E49	Row address strobe signal rise delay time ($\overline{RAS4 - 1}$)	t_{RASDR}	-	$\dfrac{t_{CYC}}{4}$ [*1] -2	-	ns
E50	Column address strobe signal fall delay time ($\overline{CAS3 - 0}$)	t_{CASDF}	-	$\dfrac{t_{CYC}}{4}$ [*1] -8	$\dfrac{t_{CYC}}{4}$ [*1] $+2$	ns
E51	Column address strobe signal rise delay time ($\overline{CAS3 - 0}$)	t_{CASDR}	-	$\dfrac{t_{CYC}}{4}$ [*1] -2	-	ns

Notes:

(1) *1 t_{CYC} is the SYSCLK cycle time.

(2) n_R and n_C change according to the DRAMCTR and MEMCTR0 setting. The correspondence is shown below.

$$n_R = 2 + 2 RTC$$

$$n_C = 4 + \frac{4}{n_f} WC$$

When FRQ = 0, $n_f = 2$, and when FRQ = 1, $n_f = 4$.

Table 13-4-7 AC Characteristics (7)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to} + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item	Symbol	Conditions	Allowable values		Unit
			Min.	Max.	
DRAM mode data transfer signal output timing (Refer to Fig. 13-4-6.)					
E52 Read data enable signal fall delay time ($\overline{RE}$)	t_{DREDF}	-	$\dfrac{t_{CYC}}{4} - 8$ *1	$\dfrac{t_{CYC}}{4} + 2$ *1	ns
E53 Read data enable signal rise delay time ($\overline{RE}$)	t_{DREDR}	-	$\dfrac{t_{CYC}}{4} - 2$ *1	-	ns
E54 Write data enable signal fall delay time ($\overline{WE3 - 0}$)	t_{DWEDF}	-	$\dfrac{t_{CYC}}{4} \times n_R - 8$ *1	$\dfrac{t_{CYC}}{4} \times n_R + 2$ *1	ns
E55 Write data enable signal rise delay time ($\overline{WE3 - 0}$)	t_{DWEDR}	-	$\dfrac{t_{CYC}}{4} - 2$ *1	-	ns
E56 Write data delay time (D31 - 0)	t_{DWDD}	-	-	$\dfrac{t_{CYC}}{4} \times n_R + 2$ *1	ns
E57 Write data hold time (D31 - 0)	t_{DWDH}	-	0	-	ns
E58 Read data setup time (D31 - 0)	t_{DRDS}	-	15	-	ns
E59 Read data hold time (D31 - 0)	t_{DRDH}	-	0	-	ns

Note:

*1 t_{CYC} is the SYSCLK cycle time.
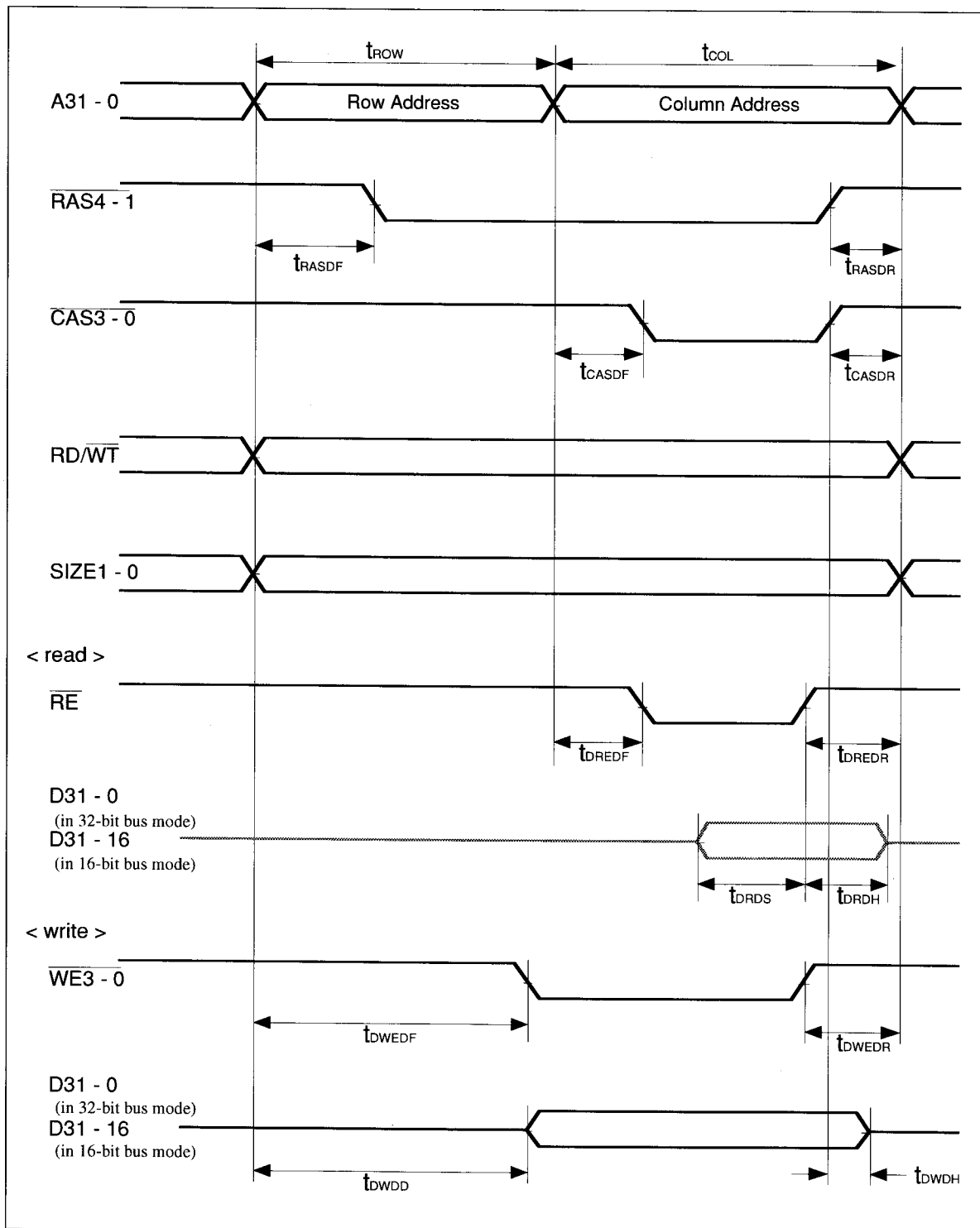
Fig. 13-4-6 DRAM Mode Data Transfer Signal Timing

Table 13-4-8 AC Characteristics (8)

$V_{DD} = 3.3\ V \pm 0.165\ V$
$V_{SS} = 0\ V$
$T_A = -20\ ^\circ C\ to\ +70\ ^\circ C$
$C_L = 50\ pF$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
DRAM page mode data transfer signal output timing (Refer to Fig. 13-4-7.)						
E60	Column address output width (A31 - 0)	t_{PCOL}	-	$\dfrac{t_{CYC}}{4} \times nc$ [1] -5	-	ns
E61	Row address strobe signal rise delay time ($\overline{RAS4 - 1}$)	t_{PRASDR}	-	$\dfrac{t_{CYC}}{4} - 2$ [1]	-	ns
E62	Column address strobe signal fall delay time ($\overline{CAS3 - 0}$)	t_{PCASDF}	-	$\dfrac{t_{CYC}}{4} - 8$ [1]	$\dfrac{t_{CYC}}{4} + 2$ [1]	ns
E63	Column address strobe signal rise delay time ($\overline{CAS3 - 0}$)	t_{PCASDR}	-	$\dfrac{t_{CYC}}{4} - 2$ [1]	-	ns
E64	Read data enable signal fall delay time ($\overline{RE}$)	t_{PDREDF}	-	$\dfrac{t_{CYC}}{4} - 8$ [1]	$\dfrac{t_{CYC}}{4} + 2$ [1]	ns
E65	Read data enable signal rise delay time ($\overline{RE}$)	t_{PDREDR}	-	$\dfrac{t_{CYC}}{4} - 2$ [1]	-	ns

Note:

[1] t_{CYC} is the SYSCLK cycle time.
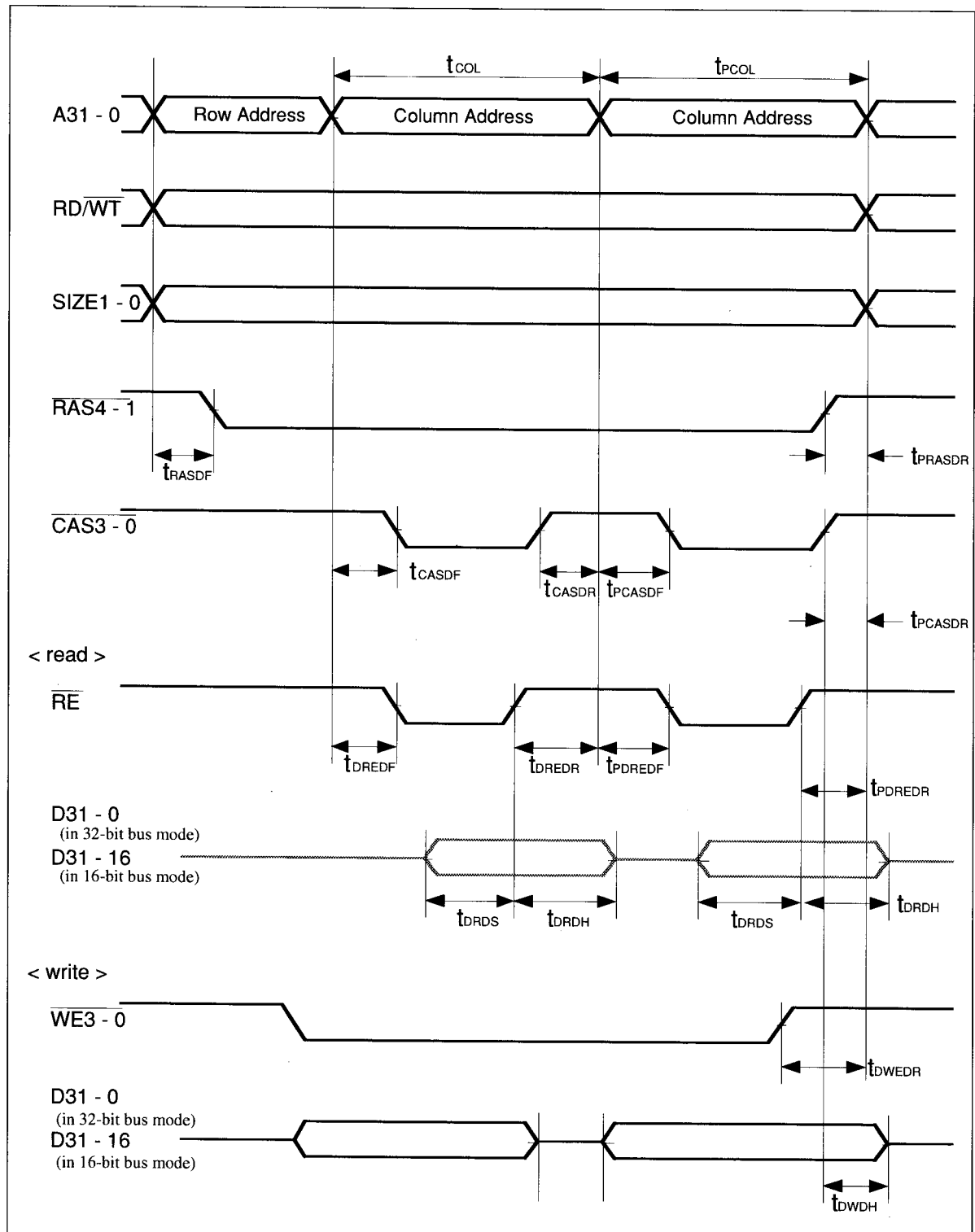
Fig. 13-4-7 DRAM Page Mode Data Transfer Signal Timing

Table 13-4-9 AC Characteristics (9)

$$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$$
$$V_{SS} = 0 \text{ V}$$
$$T_A = -20 \text{ °C to} + 70 \text{ °C}$$
$$C_L = 50 \text{ pF}$$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
DRAM refresh signal timing (Refer to Fig. 13-4-8.)						
E66	Row address strobe signal fall delay time 2 (RAS4 - 1)	t_{RASDF2}	-	$\dfrac{t_{CYC}}{2} - 2$ [*1]	-	ns
E67	Row address strobe signal rise delay time 2 (RAS4 - 1)	t_{RASDR2}	-	$\dfrac{t_{CYC}}{4} \times n_D$ [*1] $- 2$	-	ns
E68	Column address strobe signal rise delay time 2 (CAS3 - 0)	t_{CASDR2}	-	$\dfrac{t_{CYC}}{4} \times n_D$ [*1] $- 2$	-	ns

Notes:

(1) *1 t_{CYC} is the SYSCLK cycle time.

(2) n_D change according to the DRAMCTR and MEMCTR0 setting.

The correspondence is shown below.

$$n_D = 6 + \frac{8}{n_f} \text{ RERP}$$

When FRQ = 0, n_f = 2, and when FRQ = 1, n_f = 4.
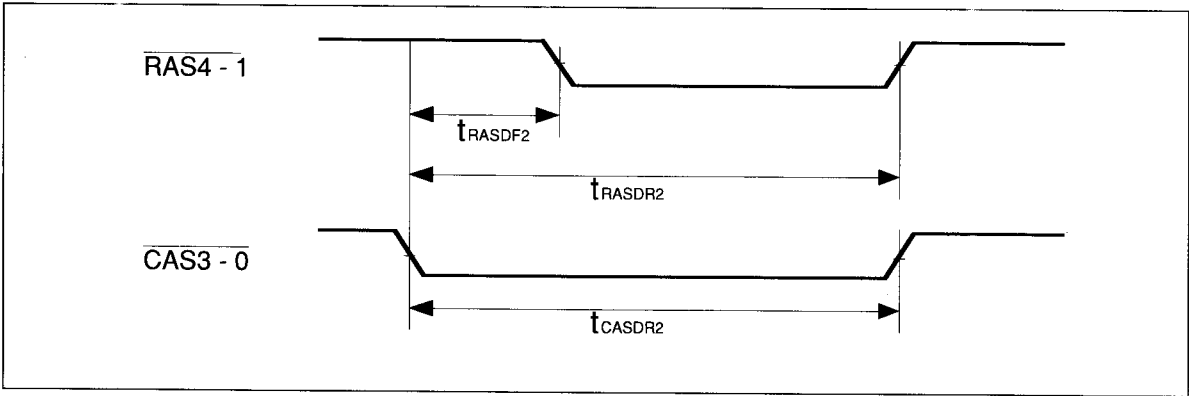
Fig. 13-4-8 DRAM Refresh Signal Timing

13.4.4 Bus Arbitration Signal Timing

Table 13-4-10 AC Characteristics (10)

$V_{DD} = 3.3\ V \pm 0.165\ V$
$V_{SS} = 0\ V$
$T_A = -20\ ^\circ C\ to + 70\ ^\circ C$
$C_L = 50\ pF$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Bus request signal timing (Refer to Figs. 13-4-9 and 13-4-10.)						
E69	Bus request signal setup time 1 ($\overline{BR}$)	t_{BRQLBS}	-	15	-	ns
E70	Bus request signal hold time 1 ($\overline{BR}$)	t_{BRQLBH}	-	0	-	ns
E71	Bus request signal setup time 2 ($\overline{BR}$)	t_{BRQLES}	-	15	-	ns
E72	Bus request signal hold time 2 ($\overline{BR}$)	t_{BRQLEH}	-	0	-	ns
E73	Bus grant signal fall delay time ($\overline{BG}$)	t_{BGDF}	-	-	5	ns
E74	Bus grant signal rise delay time ($\overline{BG}$)	t_{BGDR}	-	-	5	ns
E75	Bus tri-state delay time (A31 - 0 , D31 - 0 , $\overline{RE}$, $\overline{WE3 - 0}$, $\overline{CS7 - 0}$, $\overline{RAS4 - 1}$, $\overline{CAS3 - 0}$)	t_{BOF}	-	-	15	ns
E76	Bus buffer on delay time (A31 - 0 , D31 - 0 , $\overline{RE}$, $\overline{WE3 - 0}$, $\overline{CS7 - 0}$, $\overline{RAS4 - 1}$, $\overline{CAS3 - 0}$)	t_{BON}	-	-	15	ns
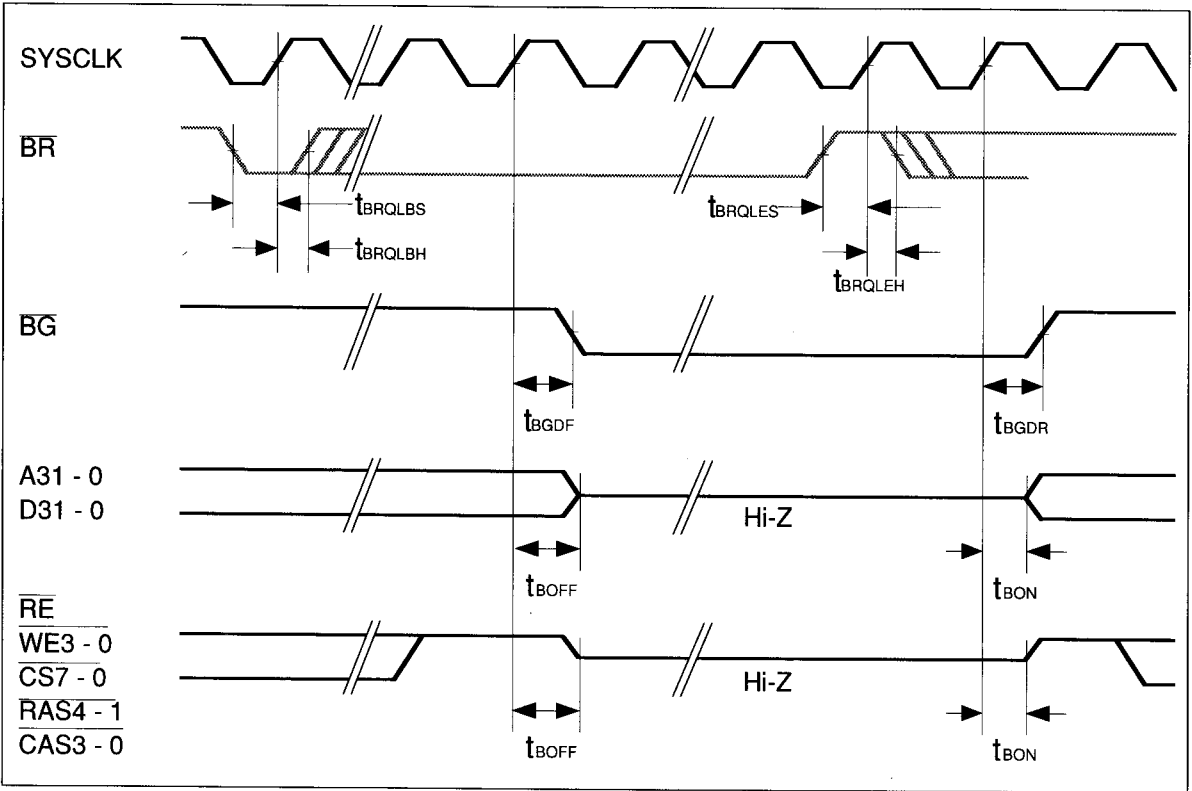
Fig. 13-4-9 Bus Request Signal Timing (1)
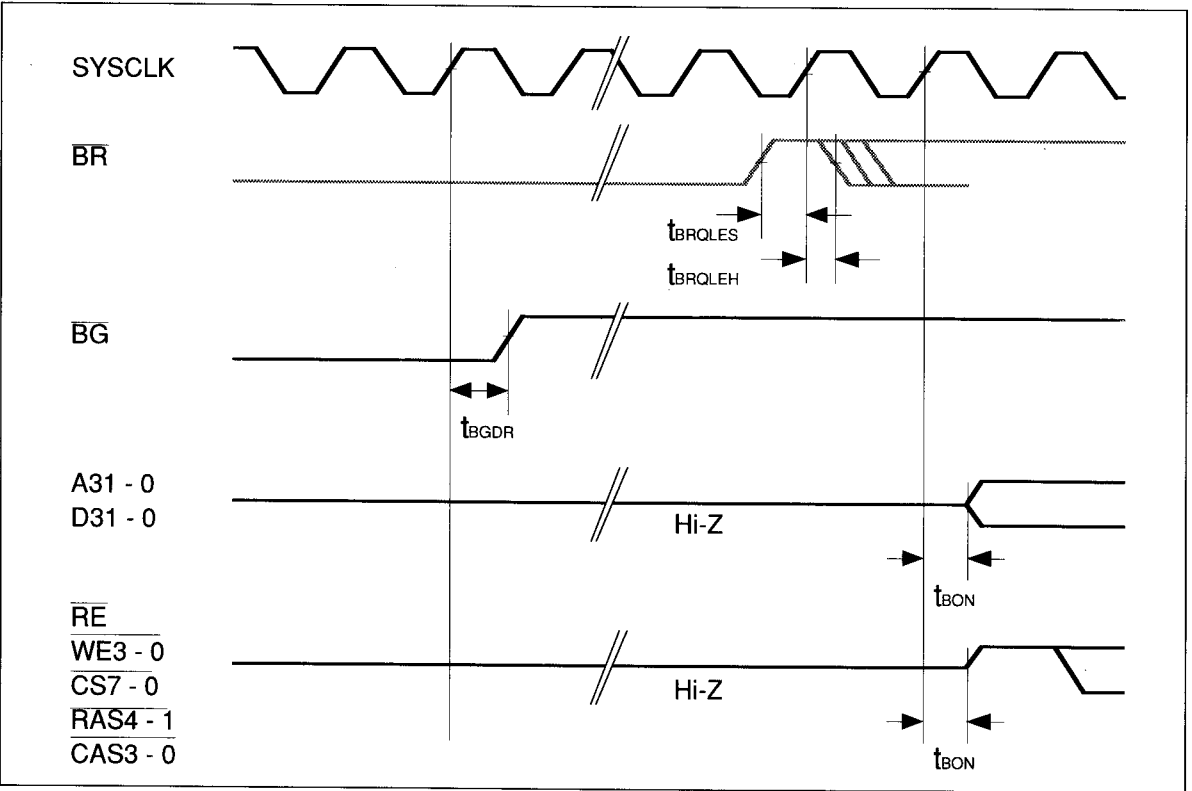
Fig. 13-4-10 Bus Request Signal Timing (2)
(Refresh request generation when bus authority is released)

13.4.5 DMA Signal Timing

Table 13-4-11 AC Characteristics (11)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to} + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item	Symbol	Conditions	Allowable values		Unit
			Min.	Max.	
DMA signal input/output timing (Refer to Fig. 13-4-1 1, Fig. 13-4-12.)					
E77 DMA request input pulse width (edge input) (DMR3 - 0)	t_{DMREW}	-	$\dfrac{t_{CYC}}{n_r}$ [*1]	-	ns
E78 DMA level transfer initiation setup time (DMK3 - 0)	t_{DMKDD}	-	-	5	ns
E79 DMA level transfer initiation hold time (DMK3 - 0)	t_{DMKUD}	-	-	5	ns

Notes:

*1 t_{CYC} is the SYSCLK cycle time.

n_r change according to the MEMCTR0 setting.

When FRQ = 0, n_r = 2, and when FRQ = 1, n_r = 4.
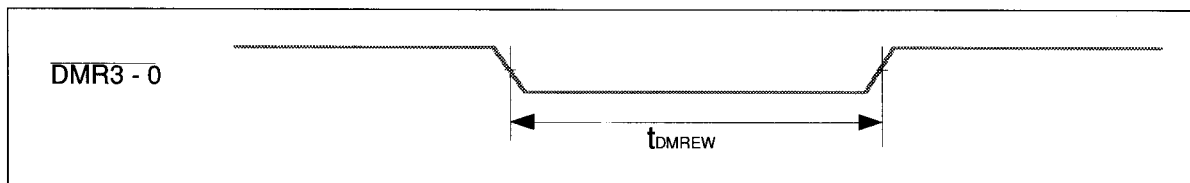
Fig. 13-4-11 DMA Request Input Timing
(in DMR edge detection mode)
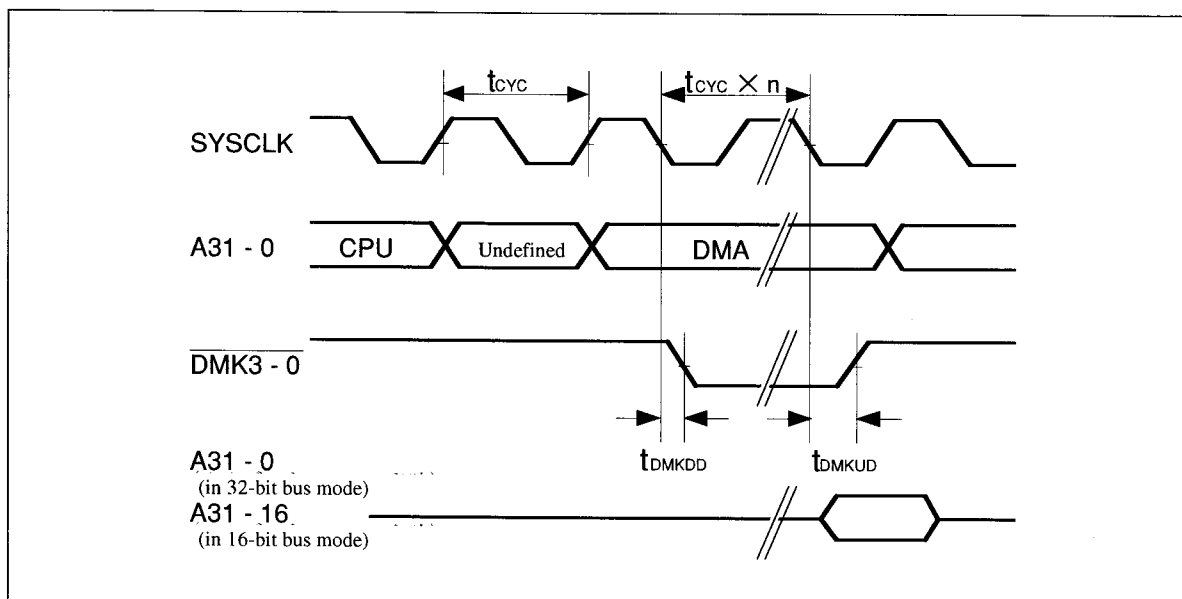
Fig. 13-4-12 DMK Output Timing
(in internal multiply by 4, bus "n" wait state synchronous mode)

13.4.6 Interrupt Signal Timing

Table 13-4-12 AC Characteristics (12)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to } + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item		Symbol	Conditions	Allowable values		単位
				Min.	Max.	
Interrupt signal input timing (Refer to Fig. 13-4-13.)						
E80	External interrupt signal pulse width ($\overline{IRQ0 - 7}$)	t_{IRQW}	FRQS pin = L level	3.5 t_{CYC}[*1]	-	ns
			FRQS pin = H level	1.5 t_{CYC}[*1]	-	ns
E81	External NMI signal pulse width ($\overline{NMIRQ}$)	t_{NMIRQW}	FRQS pin = L level	3.5 t_{CYC}[*1]	-	ns
			FRQS pin = H level	1.5 t_{CYC}[*1]	-	ns

Notes:

*1 t_{CYC} is the SYSCLK cycle time.
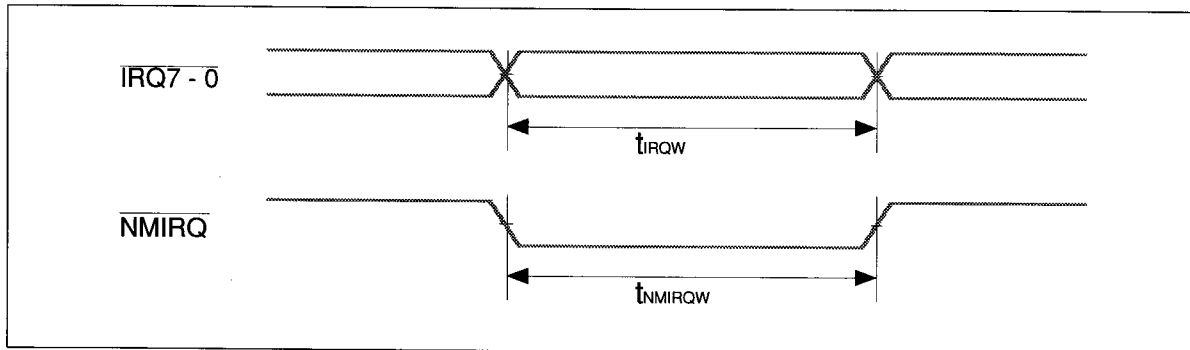
Fig. 13-4-13 Interrupt Signal Timing

13.4.7 Timer Signal Timing

Table 13-4-13 AC Characteristics (13)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to} + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Timer counter signal input/output timing (Refer to Figs. 13-4-14 and 13-4-15.)						
E82	Timer input signal setup time (TM0IO - TM5IO , TM6IOA , TM6IOB)	t_{TCIS}	-	15	-	ns
E83	Timer input signal pulse width (TM0IO - TM5IO , TM6IOA , TM6IOB)	t_{TCIW}	FRQS pin = L level	$3.5 \, t_{CYC}$ [*1]	-	ns
			FRQS pin = H level	$1.5 \, t_{CYC}$ [*1]	-	
E84	Timer counter output signal delay time (TM0IO - TM5IO , TM6IOA , TM6IOB)	t_{TCD}	-	-	5	ns

Note:

*1 t_{CYC} is the SYSCLK cycle time.
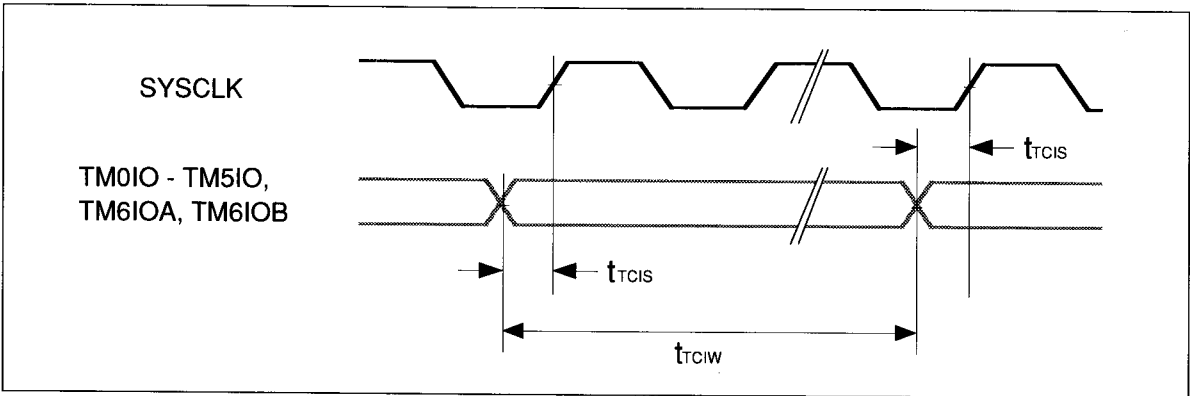
Fig. 13-4-14 Timer Counter Input Timing
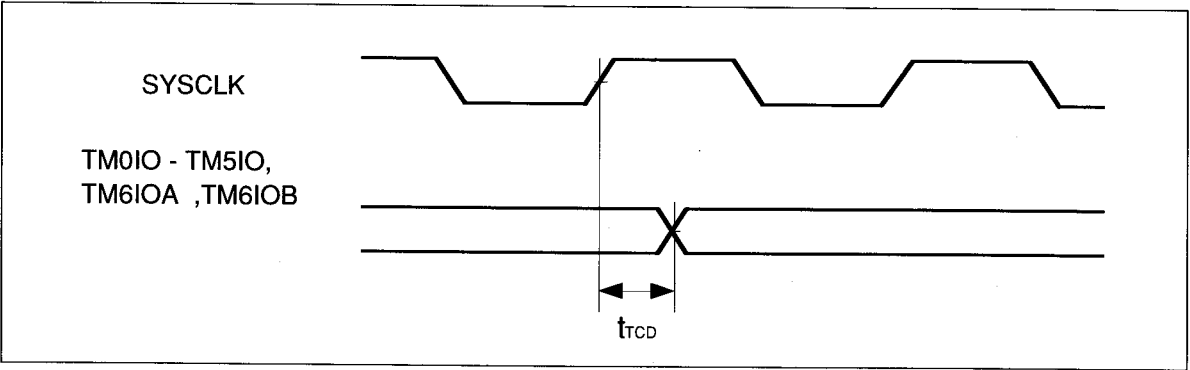
Fig. 13-4-15 Timer Counter Output Timing

13.4.8 Watchdog Timer Signal Timing

Table 13-4-14 AC Characteristics (14)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to} + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Watchdog overflow signal output timing (Refer to Fig. 13-4-16.)						
E85	Watchdog overflow signal rise delay time (WDOVF)	$t_{WDOVFRD}$	-	-	5	ns
E86	Watchdog overflow signal fall delay time (WDOVF)	$t_{WDOVFFD}$	-	-	5	ns
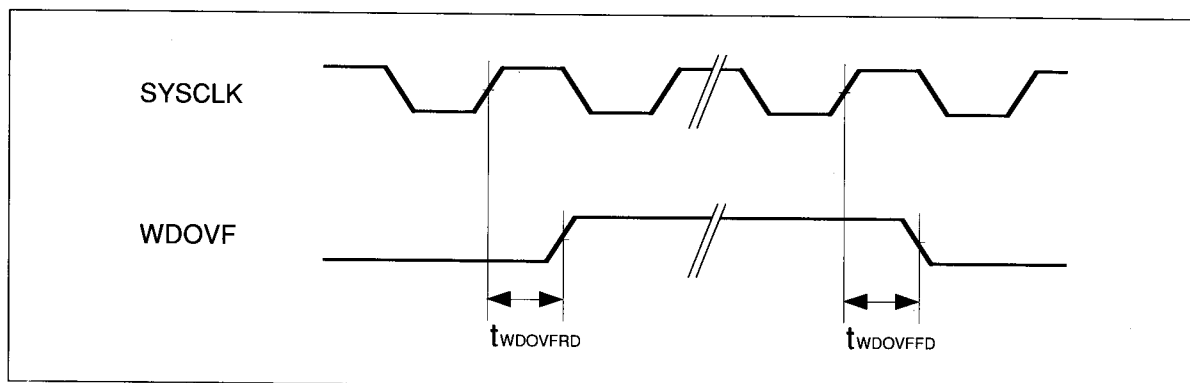
Fig. 13-4-16 Watchdog Overflow Output Timing

13.4.9 Serial Interface Signal Timing

Table 13-4-15 AC Characteristics (15)

$V_{DD} = 3.3 \text{ V} \pm 0.165 \text{ V}$
$V_{SS} = 0 \text{ V}$
$T_A = -20 \text{ °C to } + 70 \text{ °C}$
$C_L = 50 \text{ pF}$

Item		Symbol	Conditions	Allowable values		Unit
				Min.	Max.	
Serial interface signal input/output timing (Refer to Fig. 13-4-17 and Fig. 13-4-18.)						
E87	Reception data setup time (SBI2 - 0)	t_{RXDS}	-	5	-	ns
E88	Reception data hold time (SBI2 - 0)	t_{RXDH}	-	2	-	ns
E89	Transfer data delay time 1 (SBO2 - 0)	t_{TXDD1}	-	-	8	ns
E90	Transfer data delay time 2 (SBO2 - 0)	t_{TXDD2}	-	0	-	ns
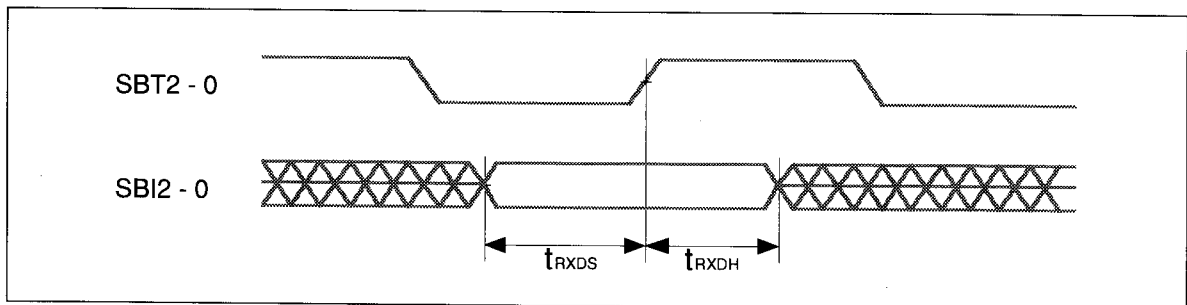
Fig. 13-4-17 Serial Interface Signal Timing (1)
(during synchronous serial reception)
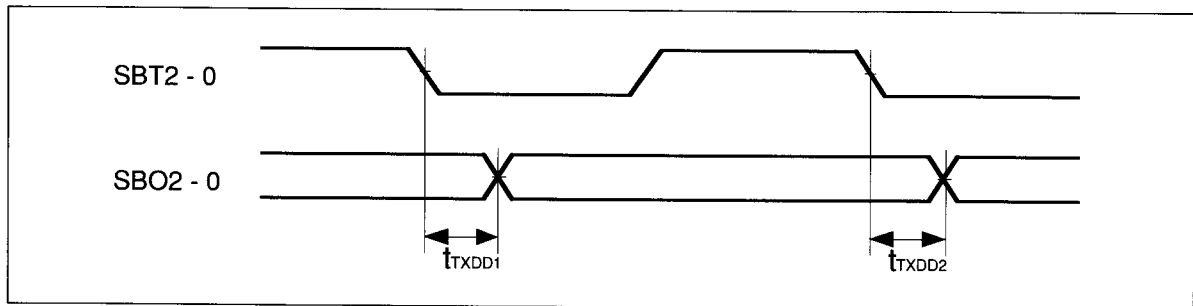
Fig. 13-4-18 Serial Interface Timing (1)
(during synchronous serial reception)

13.4.10 AC Characteristics Test Conditions

OSCI

V_{IH} : Input voltage high level
V_{IL} : Input voltage low level

SYSCLK

V_{OH} : Output voltage high level
V_{OL} : Output voltage low level

V_{IH} : $V_{IH1}(MIN),V_{IH3}(MIN),V_{IH4}(MIN)$
V_{IL} : $V_{IL1}(MAX),V_{IL3}(MAX),V_{IL4}(MAX)$

V_{OH} : V_{DD} x 0.7
V_{OL} : V_{DD} x 0.3

Input signal

V_T : Input voltage transition level

Output signal

V_T : Output voltage transition level

V_T : V_{DD} x 0.5
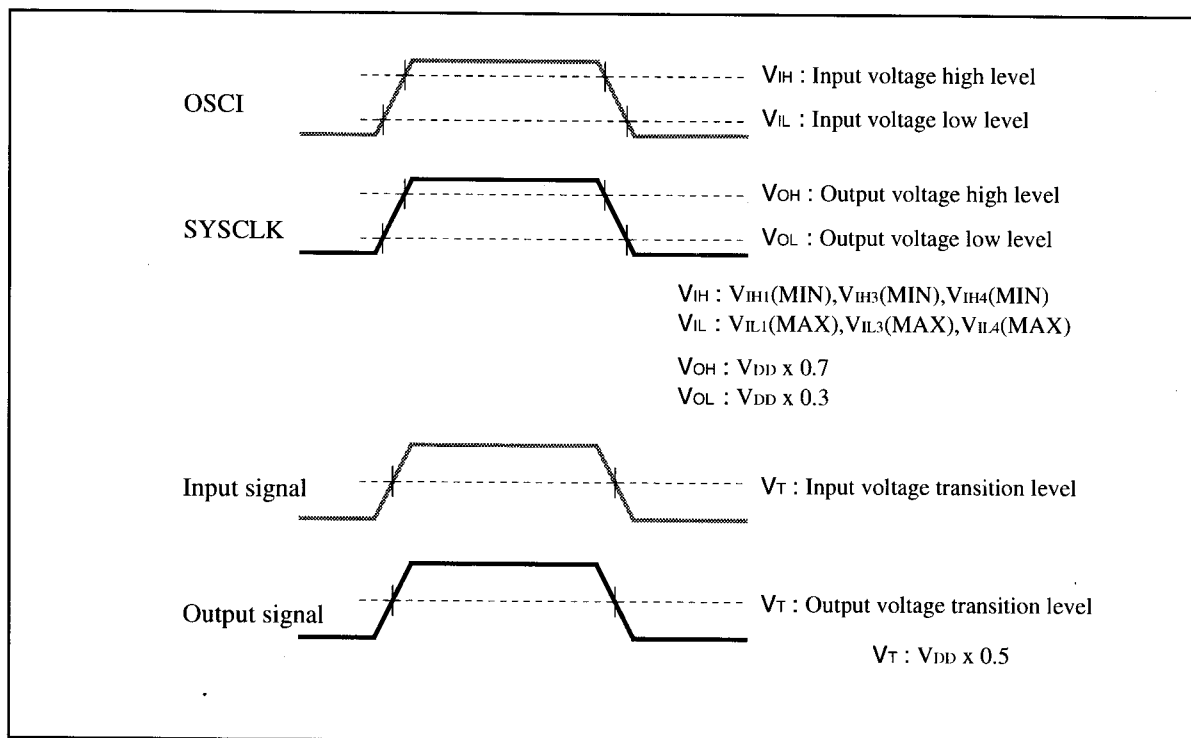
Fig. 13-4-19 AC Timing Test Voltage Levels

Appendix

Appendix A: Treatment of Pins

Treat pins as following Table A-1

Table A-1 Treatment of Pins

Name of Pins	Condition		Treatment
A31/CS7, A30, A29, A26/CS6, A27/CS5, A26/CS4/RAS4, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, RAS3/CS3, RAS2/CS2, RAS1/CS1, CS0, RE, WE3, WE2, WE1, WE0, CAS3, CAS2, CAS1, CAS0, RD/WT, SIZE1, SIZE0, D31, D30, D29, D28, D27, D26, D25, D24, D23, D22, D21, D20, D19, D18, D17, D16	When not connected with signal line		Connect to V_{DD} through resister
	When connected	When external devices and CPU have might to use buses	Connect to V_{DD} through resister
		When only CPU has might to use buses	No treatment to pullup/pulldown
D15/P17, D14/P16, D13/P15, D12/P14, D11/P13, D10/P12, D9/P11, D8/P10, D7/P07, D6/P06, D5/P05, D4/P04, D3/P03, D2/P02, D1/P01, D0/P00	When not connected with signal line		Set as port input, connect to V_{DD} through resister or set as port output, leave as floating pin
	When connected	When external devices and CPU have might to use buses	Connect to V_{DD} through resister
		When only CPU has might to use buses	No treatment to pullup/pulldown
TM6IOB/P27, TM6IOA/P26, TM5IO/P25, TM4IO/P24, TM3IO/P23, TM2IO/P22, TM1IO/SBT1/P21, TM0IO/SBT0/P20, SBO1/P31, SBO0/P30	When not connected with signal line		Set as port input, connect to V_{DD} through register or set as port output, leave as floating pin
	Connected with signal line		No treatment to pullup/pulldown
IRQ7, IRQ6, IRQ5, IRQ4, IRQ3, IRQ2, IRQ1, IRQ0, NMIRQ, DK, BR, DMR3, DMR2, DMR1, DMR0, CTS, SBT2, SBI2, SBI1, SBI0	When not connected with signal line		Connect to V_{DD} through resister
	Connected with signal line		No treatment to pullup/pulldown
DSCLK, DSDAT			Connect to Vss through resister If use serial debugger, recommended value for resister is 1 kΩ
BG, WDOVF, OSCO, SYSCLK, DMK3, DMK2, DMK1, DMK0, SBO2			No treatment to pullup/pulldown
FRQS			Select pullup/pulldown Refer to 3.4.1
OSCI			Connect oscillator Refer to 3.3
BMODE			Select pullup/pulldown Refer to 5.5
MMODE			Connect to Vss through resister

Appendix B: External Pin Statuses

Table B-1 shows the operating status of external pins concerning the BC.

Table B-1 Operating Status of External Pins Concerning BC

Operating status Pin	Reset	STOP mode	SLEEP mode	HALT mode	Bus release mode
SYSCLK	L	L	Operating	L	Operating
A31-0	x'9C000000 [*1]	Hold	Operating	Hold	Hi-Z
D31~0	Hi-Z	Hi-Z	Operating	Hi-Z	Hi-Z
RAS4~1	H	H	Operating	H	Hi-Z
CAS3~0	H	H	Operating	H	Hi-Z
CS7~0	H	H	Operating	H	Hi-Z
RE	H	H	Operating	H	Hi-Z
WE3~0	H	H	Operating	H	Hi-Z
DK [*2]	H	H	Operating	H	Hi-Z
RD/WT	H	Hold	Operating	Hold	Hi-Z
SIZE1~0	H	Hold	Operating	Hold	Hi-Z
BG	H	H	Operating	H	L

Hi-Z: High impedance

Hold: Hold status from immediately preceding external bus cycle

Input: Input status

L: Low-level output

H: High-level output

[*1]   Because A31 and CS7 as well as A28 through 26 and CS6 through 4 are dual-purpose pins, "x'9C000000" are output.

[*2]   Shows the input signal level. Adjust so that the prescribed level is attained.

Appendix C: List of Register Maps

Table C-1 Register Map (1)

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'2000000X			IVAR3				IVAR2				IVAR1				IVAR0		Interrupt vector
x'2000001X							IVAR6				IVAR5				IVAR4		
x'2000004X															CPUM		CPU mode
x'2000007X															CHCTR		Cache control
x'2800000X	Way 0 entry 0 offset 3				Way 0 entry 0 offset 2				Way 0 entry 0 offset 1				Way 0 entry 0 offset 0				Cache data
x'2800001X	Way 0 entry 1 offset 3				Way 0 entry 1 offset 2				Way 0 entry 1 offset 1				Way 0 entry 1 offset 0				
~																	
x'280007EX	Way 0 entry 126 offset 3				Way 0 entry 126 offset 2				Way 0 entry 126 offset 1				Way 0 entry 126 offset 0				
x'280007FX	Way 0 entry 127 offset 3				Way 0 entry 127 offset 2				Way 0 entry 127 offset 1				Way 0 entry 127 offset 0				
x'2800100X	Way 1 entry 0 offset 3				Way 1 entry 0 offset 2				Way 1 entry 0 offset 1				Way 1 entry 0 offset 0				
x'2800101X	Way 1 entry 1 offset 3				Way 1 entry 1 offset 2				Way 1 entry 1 offset 1				Way 1 entry 1 offset 0				
~																	
x'280017EX	Way 1 entry 126 offset 3				Way 1 entry 126 offset 2				Way 1 entry 126 offset 1				Way 1 entry 126 offset 0				
x'280017FX	Way 1 entry 127 offset 3				Way 1 entry 127 offset 2				Way 1 entry 127 offset 1				Way 1 entry 127 offset 0				
x'2810000X														Way 0 entry 0			
x'2810001X															Way 0 entry 1		
~																	
x'281007EX														Way 0 entry 126			
x'281007FX															Way 0 entry 127		

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Cache data
x'2810100X															Way 1 entry 0		
x'2810101X															Way 1 entry 1		
~																	
x'281017EX															Way 1 entry 126		
x'281017FX															Way 1 entry 127		
x'2820000X		Way 0 entry 0 offset 3				Way 0 entry 0 offset 2				Way 0 entry 0 offset 1				Way 0 entry 0 offset 0			
x'2820001X		Way 0 entry 1 offset 3				Way 0 entry 1 offset 2				Way 0 entry 1 offset 1				Way 0 entry 1 offset 0			
~																	
x'282007EX		Way 0 entry 126 offset 3				Way 0 entry 126 offset 2				Way 0 entry 126 offset 1				Way 0 entry 126 offset 0			
x'282007FX		Way 0 entry 127 offset 3				Way 0 entry 127 offset 2				Way 0 entry 127 offset 1				Way 0 entry 127 offset 0			
x'2820100X		Way 1 entry 0 offset 3				Way 1 entry 0 offset 2				Way 1 entry 0 offset 1				Way 1 entry 0 offset 0			
x'2820101X		Way 1 entry 1 offset 3				Way 1 entry 1 offset 2				Way 1 entry 1 offset 1				Way 1 entry 1 offset 0			
~																	
x'282017EX		Way 1 entry 126 offset 3				Way 1 entry 126 offset 2				Way 1 entry 126 offset 1				Way 1 entry 126 offset 0			
x'282017FX		Way 1 entry 127 offset 3				Way 1 entry 127 offset 2				Way 1 entry 127 offset 1				Way 1 entry 127 offset 0			
x'2830000X															Way 0 entry 0		
x'2830001X															Way 0 entry 1		
~																	
x'283007EX															Way 0 entry 126		
x'283007FX															Way 0 entry 127		

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'2830100X													Way 1 entry 0				Cache data
x'2830101X													Way 1 entry 1				
~																	
x'283017EX													Way 1 entry 126				
x'283017FX													Way 1 entry 127				
x'2840000X													Way 0 entry 0				
x'2840001X													Way 0 entry 1				
~																	
x'284007EX													Way 0 entry 126				
x'284007FX													Way 0 entry 127				
x'2840100X													Way 1 entry 0				
x'2840101X													Way 1 entry 1				
~																	
x'284017EX													Way 1 entry 126				
x'284017FX													Way 1 entry 127				
x'3200001X															IOBCTR		I/O bus control
x'3200002X	MEMCTR7		MEMCTR6		MEMCTR5		MEMCTR4		MEMCTR3		MEMCTR2		MEMCTR1		MEMCTR0		Memory control
x'3200004X													REFCNT		DRAMCTR		

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'3200010X		DM0 CYC	DM0CNT			DM0DST				DM0SRC				DM0CTR			DMA
x'3200020X		DM1 CYC	DM1CNT			DM1DST				DM1SRC				DM1CTR			
x'3200040X		DM2 CYC	DM2CNT			DM2DST				DM2SRC				DM2CTR			
x'3200080X		DM3 CYC	DM3CNT			DM3DST				DM3SRC				DM3CTR			
x'3400010X				G3ICR				G2ICR								NMICR	Interrupt control
x'3400011X				G7ICR				G6ICR				G5ICR				G4ICR	
x'3400012X				G11ICR				G10ICR				G9ICR				G8ICR	
x'3400013X				G15ICR				G14ICR				G13ICR				G12ICR	
x'3400014X				G19ICR				G18ICR				G17ICR				G16ICR	
x'3400015X				G23ICR				G22ICR				G21ICR				G20ICR	
x'3400016X				G27ICR				G26ICR				G25ICR				G24ICR	
x'3400017X								G30ICR				G29ICR				G28ICR	
x'3400020X																IAGR	
x'3400028X																EXTMD	
x'3400080X			SC0STR				SC0 RXB	SC0 TXB				SC0 ICR				SC0CTR	Serial interface
x'3400081X			SC1STR				SC1 RXB	SC1 TXB				SC1 ICR				SC1CTR	
x'3400082X		SC2 TIM	SC2 STR				SC2 RXB	SC2 TXB				SC2 ICR				SC2CTR	

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
x'3400100X	TM0 MD	TM1 MD	TM2 MD	TM3 MD													8-bit timer
x'3400101X	TM0 BR	TM1 BR	TM2 BR	TM3 BR													
x'3400102X	TM0 BC	TM1 BC	TM2 BC	TM3 BC													
x'3400108X	TM4 MD		TM5 MD		TM6MD												16-bit timer
x'3400109X	TM4BR		TM5BR														
x'340010AX	TM4BC		TM5BC		TM6BC												
x'340010BX					TM6 MDA	TM6 MDB											
x'340010CX					TM6CA												
x'340010DX					TM6CB												
x'3400400X	WDBC		WD CTR		RST CTR												Watchdog timer
x'3600800X	P0OUT	P1OUT			P2OUT	P3OUT											I/O port
x'3600801X																	
x'3600802X	P0MD	P1MD			P2MD	P3MD											
x'3600803X																	
x'3600804X					P2SS				P4SS								
x'3600805X																	
x'3600806X	P0DIR	P1DIR			P2DIR	P3DIR											
x'3600807X																	
x'3600808X	P0IN	P1IN			P2IN	P3IN											

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

Appendix D: Instruction Set

List of Instructions (Code Length, Number of Cycles *)

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
MOV	MOV	Dm	Dn	S0	1	1	
	MOV	Dm	An	D0	2	1	
	MOV	Am	Dn	D0	2	1	
	MOV	Am	An	S0	1	1	
	MOV	SP	An	S0	1	1	
	MOV	Am	SP	D0	2	1	
	MOV	PSW	Dn	D0	2	1	
	MOV	Dm	PSW	D0	2	1	
	MOV	MDR	Dn	D0	2	1	
	MOV	Dm	MDR	D0	2	1	
	MOV	(Am)	Dn	S0	1	1	
	MOV	(d8,Am)	Dn	D1	3	1	
	MOV	(d16,Am)	Dn	D2	4	1	
	MOV	(d32,Am)	Dn	D4	6	2	
	MOV	(d8,SP)	Dn	S1	2	1	
	MOV	(d16,SP)	Dn	D2	4	1	
	MOV	(d32,SP)	Dn	D4	6	2	
	MOV	(Di,Am)	Dn	D0	2	1	
	MOV	(abs16)	Dn	S2	3	1	
	MOV	(abs32)	Dn	D4	6	2	
	MOV	(Am)	An	D0	2	1	
	MOV	(d8,Am)	An	D1	3	1	
	MOV	(d16,Am)	An	D2	4	1	
	MOV	(d32,Am)	An	D4	6	2	
	MOV	(d8,SP)	An	S1	2	1	
	MOV	(d16,SP)	An	D2	4	1	
	MOV	(d32,SP)	An	D4	6	2	
	MOV	(Di,Am)	An	D0	2	1	
	MOV	(abs16)	An	D2	4	1	
	MOV	(abs32)	An	D4	6	2	
	MOV	(d8,Am)	SP	D1	3	1	
	MOV	Dm	(An)	S0	1	1	
	MOV	Dm	(d8,An)	D1	3	1	
	MOV	Dm	(d16,An)	D2	4	1	
	MOV	Dm	(d32,An)	D4	6	2	
	MOV	Dm	(d8,SP)	S1	2	1	
	MOV	Dm	(d16,SP)	D2	4	1	
	MOV	Dm	(d32,SP)	D4	6	2	
	MOV	Dm	(Di,An)	D0	2	2	
	MOV	Dm	(abs16)	S2	3	1	
	MOV	Dm	(abs32)	D4	6	2	
	MOV	Am	(An)	D0	2	1	
	MOV	Am	(d8,An)	D1	3	1	
	MOV	Am	(d16,An)	D2	4	1	
	MOV	Am	(d32,An)	D4	6	2	
	MOV	Am	(d8,SP)	S1	2	1	
	MOV	Am	(d16,SP)	D2	4	1	
	MOV	Am	(d32,SP)	D4	6	2	
	MOV	Am	(Di,An)	D0	2	2	
	MOV	Am	(abs16)	D2	4	1	
	MOV	Am	(abs32)	D4	6	2	
	MOV	SP	(d8,An)	D1	3	1	
	MOV	imm8	Dn	S1	2	1	
	MOV	imm16	Dn	S2	3	1	
	MOV	imm32	Dn	D4	6	2	
	MOV	imm8	An	S1	2	1	
	MOV	imm16	An	S2	3	1	
	MOV	imm32	An	D4	6	2	

* The number of execution cycles was calculated under the following conditions.
   (1) No pipeline extension (2) 2 cycles for the instruction fetch, 1 cycle for the load/store

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
MOVBU	MOVBU	(Am)	Dn	D0	2	1	
	MOVBU	(d8,Am)	Dn	D1	3	1	
	MOVBU	(d16,Am)	Dn	D2	4	1	
	MOVBU	(d32,Am)	Dn	D4	6	2	
	MOVBU	(d8,SP)	Dn	D1	3	1	
	MOVBU	(d16,SP)	Dn	D2	4	1	
	MOVBU	(d32,SP)	Dn	D4	6	2	
	MOVBU	(Di,Am)	Dn	D0	2	1	
	MOVBU	(abs16)	Dn	S2	3	1	
	MOVBU	(abs32)	Dn	D4	6	2	
	MOVBU	Dm	(An)	D0	2	1	
	MOVBU	Dm	(d8,An)	D1	3	1	
	MOVBU	Dm	(d16,An)	D2	4	1	
	MOVBU	Dm	(d32,An)	D4	6	2	
	MOVBU	Dm	(d8,SP)	D1	3	1	
	MOVBU	Dm	(d16,SP)	D2	4	1	
	MOVBU	Dm	(d32,SP)	D4	6	2	
	MOVBU	Dm	(Di,An)	D0	2	2	
	MOVBU	Dm	(abs16)	S2	3	1	
	MOVBU	Dm	(abs32)	D4	6	2	
MOVHU	MOVHU	(Am)	Dn	D0	2	1	
	MOVHU	(d8,Am)	Dn	D1	3	1	
	MOVHU	(d16,Am)	Dn	D2	4	1	
	MOVHU	(d32,Am)	Dn	D4	6	2	
	MOVHU	(d8,SP)	Dn	D1	3	1	
	MOVHU	(d16,SP)	Dn	D2	4	1	
	MOVHU	(d32,SP)	Dn	D4	6	2	
	MOVHU	(Di,Am)	Dn	D0	2	1	
	MOVHU	(abs16)	Dn	S2	3	1	
	MOVHU	(abs32)	Dn	D4	6	2	
	MOVHU	Dm	(An)	D0	2	1	
	MOVHU	Dm	(d8,An)	D1	3	1	
	MOVHU	Dm	(d16,An)	D2	4	1	
	MOVHU	Dm	(d32,An)	D4	6	2	
	MOVHU	Dm	(d8,SP)	D1	3	1	
	MOVHU	Dm	(d16,SP)	D2	4	1	
	MOVHU	Dm	(d32,SP)	D4	6	2	
	MOVHU	Dm	(Di,An)	D0	2	2	
	MOVHU	Dm	(abs16)	S2	3	1	
	MOVHU	Dm	(abs32)	D4	6	2	
EXT	EXT		Dn	D0	2	1	
EXTB	EXTB		Dn	S0	1	1	
EXTBU	EXTBU		Dn	S0	1	1	
EXTH	EXTH		Dn	S0	1	1	
EXTHU	EXTHU		Dn	S0	1	1	
MOVM	MOVM	(SP)	regs	S1	2	1	Number of registers with regs specification = 0
						2	Number of registers with regs specification = 1
						3	Number of registers with regs specification = 2
						4	Number of registers with regs specification = 3
						5	Number of registers with regs specification = 4
						8	Number of registers with regs specification = 7
						9	Number of registers with regs specification = 8
						10	Number of registers with regs specification = 9
						11	Number of registers with regs specification = 10
						12	Number of registers with regs specification = 11
	MOVM	regs	(SP)	S1	2	1	Number of registers with regs specification = 0
						1	Number of registers with regs specification = 1
						2	Number of registers with regs specification = 2
						3	Number of registers with regs specification = 3

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
MOVM						4	Number of registers with regs specification = 4
						8	Number of registers with regs specification = 7
						9	Number of registers with regs specification = 8
						10	Number of registers with regs specification = 9
						11	Number of registers with regs specification = 10
						12	Number of registers with regs specification = 11
CLR	CLR		Dn	S0	1	1	
ADD	ADD	Dm	Dn	S0	1	1	
	ADD	Dm	An	D0	2	1	
	ADD	Am	Dn	D0	2	1	
	ADD	Am	An	D0	2	1	
	ADD	imm8	Dn	S1	2	1	
	ADD	imm16	Dn	D2	4	1	
	ADD	imm32	Dn	D4	6	2	
	ADD	imm8	An	S1	2	1	
	ADD	imm16	An	D2	4	1	
	ADD	imm32	An	D4	6	2	
	ADD	imm8	SP	D1	3	1	
	ADD	imm16	SP	D2	4	1	
	ADD	imm32	SP	D4	6	2	
ADDC	ADDC	Dm	Dn	D0	2	1	
SUB	SUB	Dm	Dn	D0	2	1	
	SUB	Dm	An	D0	2	1	
	SUB	Am	Dn	D0	2	1	
	SUB	Am	An	D0	2	1	
	SUB	imm32	Dn	D4	6	2	
	SUB	imm32	An	D4	6	2	
SUBC	SUBC	Dm	Dn	D0	2	1	
MUL	MUL	Dm	Dn	D0	2	3	Dn=0
						13	Dn is a value that can be expressed by 1 byte
						21	Dn is a value that can be expressed by 2 byte
						29	Dn is a value that can be expressed by 3 byte
						34	Dn is a value that can be expressed by 4 byte
MULU	MULU	Dm	Dn	D0	2	3	Dn=0
						13	Dn is a value that can be expressed by 1 byte
						21	Dn is a value that can be expressed by 2 byte
						29	Dn is a value that can be expressed by 3 byte
						34	Dn is a value that can be expressed by 4 byte
DIV	DIV	Dm	Dn	D0	2	4	{MDR,Dn}=0
						14	{MDR, Dn} is a value that can be expressed by 1 byte
						22	{MDR, Dn} is a value that can be expressed by 2 byte
						30	{MDR, Dn} is a value that can be expressed by 3 byte
						38	{MDR, Dn} is a value that can be expressed by 4 bytes or more
DIVU	DIVU	Dm	Dn	D0	2	4	{MDR,Dn}=0
						14	{MDR, Dn} is a value that can be expressed by 1 byte
						22	{MDR, Dn} is a value that can be expressed by 2 byte
						30	{MDR, Dn} is a value that can be expressed by 3 byte
						38	{MDR, Dn} is a value that can be expressed by 4 bytes or more
INC	INC		Dn	S0	1	1	
	INC		An	S0	1	1	
INC4	INC4		An	S0	1	1	
CMP	CMP	Dm	Dn	S0	1	1	
	CMP	Dm	An	D0	2	1	
	CMP	Am	Dn	D0	2	1	
	CMP	Am	An	S0	1	1	
	CMP	imm8	Dn	S1	2	1	
	CMP	imm16	Dn	D2	4	1	
	CMP	imm32	Dn	D4	6	2	
	CMP	imm8	An	S1	2	1	
	CMP	imm16	An	D2	4	1	
	CMP	imm32	An	D4	6	2	

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
AND	AND	Dm	Dn	D0	2	1	
	AND	imm8	Dn	D1	3	1	
	AND	imm16	Dn	D2	4	1	
	AND	imm32	Dn	D4	6	2	
	AND	imm16	PSW	D2	4	1	
OR	OR	Dm	Dn	D0	2	1	
	OR	imm8	Dn	D1	3	1	
	OR	imm16	Dn	D2	4	1	
	OR	imm32	Dn	D4	6	2	
	OR	imm16	PSW	D2	4	1	
XOR	XOR	Dm	Dn	D0	2	1	
	XOR	imm16	Dn	D2	4	1	
	XOR	imm32	Dn	D4	6	2	
NOT	NOT		Dn	D0	2	1	
BTST	BTST	imm8	Dn	D1	3	1	
	BTST	imm16	Dn	D2	4	1	
	BTST	imm32	Dn	D4	6	2	
	BTST	imm8	(d8,An)	D2	4	4	
	BTST	imm8	(abs32)	D5	7	5	
BSET	BSET	Dm	(An)	D0	2	5	
	BSET	imm8	(d8,An)	D2	4	5	
	BSET	imm8	(abs32)	D5	7	6	
BCLR	BCLR	Dm	(An)	D0	2	5	
	BCLR	imm8	(d8,An)	D2	4	5	
	BCLR	imm8	(abs32)	D5	7	6	
ASR	ASR	Dm	Dn	D0	2	1	
	ASR	imm8	Dn	D1	3	1	
LSR	LSR	Dm	Dn	D0	2	1	
	LSR	imm8	Dn	D1	3	1	
ASL	ASL	Dm	Dn	D0	2	1	
	ASL	imm8	Dn	D1	3	1	
ASL2	ASL2		Dn	S0	1	1	
ROR	ROR		Dn	D0	2	1	
ROL	ROL		Dn	D0	2	1	
Bcc	BEQ	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BNE	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BGT	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BGE	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BLE	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BLT	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BHI	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BCC	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BLS	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BCS	(d8,PC)		S1	2	3 / 1*	Branch condition met/not met
	BVC	(d8,PC)		D1	3	4 / 2*	Branch condition met/not met
	BVS	(d8,PC)		D1	3	4 / 2*	Branch condition met/not met
	BNC	(d8,PC)		D1	3	4 / 2*	Branch condition met/not met
	BNS	(d8,PC)		D1	3	4 / 2*	Branch condition met/not met
	BRA	(d8,PC)		S1	2	3	Branch condition met
Lcc	LEQ			S0	1	1 / 2*	Branch condition met/not met
	LNE			S0	1	1 / 2*	Branch condition met/not met
	LGT			S0	1	1 / 2*	Branch condition met/not met
	LGE			S0	1	1 / 2*	Branch condition met/not met
	LLE			S0	1	1 / 2*	Branch condition met/not met
	LLT			S0	1	1 / 2*	Branch condition met/not met
	LHI			S0	1	1 / 2*	Branch condition met/not met
	LCC			S0	1	1 / 2*	Branch condition met/not met
	LLS			S0	1	1 / 2*	Branch condition met/not met
	LCS			S0	1	1 / 2*	Branch condition met/not met
	LRA			S0	1	1	Branch condition met

* Varies according to the state of the instruction buffer.

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
SETLB	SETLB			S0	1	1	
JMP	JMP	(An)		D0	2	3	
	JMP	(d16,PC)		S2	3	2	
	JMP	(d32,PC)		S4	5	4	
CALL	CALL	(d16,PC)	regs,imm8	S4	5	2	Number of registers with regs specification = 0
						3	Number of registers with regs specification = 1
						4	Number of registers with regs specification = 2
						5	Number of registers with regs specification = 3
						6	Number of registers with regs specification = 4
						9	Number of registers with regs specification = 7
						10	Number of registers with regs specification = 8
						11	Number of registers with regs specification = 9
						12	Number of registers with regs specification = 10
						13	Number of registers with regs specification = 11
CALL	CALL	(d32,PC)	regs,imm8	S6	7	5	Number of registers with regs specification = 0
						5	Number of registers with regs specification = 1
						6	Number of registers with regs specification = 2
						7	Number of registers with regs specification = 3
						8	Number of registers with regs specification = 4
						11	Number of registers with regs specification = 7
						12	Number of registers with regs specification = 8
						13	Number of registers with regs specification = 9
						14	Number of registers with regs specification = 10
						15	Number of registers with regs specification = 11
CALLS	CALLS	(An)		D0	2	3	
	CALLS	(d16,PC)		D2	4	3	
	CALLS	(d32,PC)		D4	6	4	
RET	RET	regs,imm8		S2	3	6	Number of registers with regs specification = 0
						6	Number of registers with regs specification = 1
						6	Number of registers with regs specification = 2
						6	Number of registers with regs specification = 3
						7	Number of registers with regs specification = 4
						10	Number of registers with regs specification = 7
						11	Number of registers with regs specification = 8
						12	Number of registers with regs specification = 9
						13	Number of registers with regs specification = 10
						14	Number of registers with regs specification = 11
RETF	RETF	regs,imm8		S2	3	2	Number of registers with regs specification = 0
						2	Number of registers with regs specification = 1
						3	Number of registers with regs specification = 2
						4	Number of registers with regs specification = 3
						5	Number of registers with regs specification = 4
						8	Number of registers with regs specification = 7
						9	Number of registers with regs specification = 8
						10	Number of registers with regs specification = 9
						11	Number of registers with regs specification = 10
						12	Number of registers with regs specification = 11
RETS	RETS			D0	2	6	
RTI	RTI			D0	2	4	
TRAP	TRAP			D0	2	4	
NOP	NOP			S0	1	1	

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
UDF	UDF00~15	Dm	Dn	D0	2	User-defined	
	UDF00~15	imm8	Dn	D1	3	User-defined	
	UDF00~15	imm16	Dn	D2	4	User-defined	
	UDF00~15	imm32	Dn	D4	6	User-defined	
	UDFU00~15	imm8	Dn	D1	3	User-defined	
	UDFU00~15	imm16	Dn	D2	4	User-defined	
	UDFU00~15	imm32	Dn	D4	6	User-defined	
	UDF20~35	Dm	Dn	D0	2	User-defined	
	UDF20~35	imm8	Dn	D1	3	User-defined	
	UDF20~35	imm16	Dn	D2	4	User-defined	
	UDF20~35	imm32	Dn	D4	6	User-defined	
	UDFU20~35	imm8	Dn	D1	3	User-defined	
	UDFU20~35	imm16	Dn	D2	4	User-defined	
	UDFU20~35	imm32	Dn	D4	6	User-defined	

Appendix E. Extension Instruction Specifications

Operation Extension Functions

The MN1030 Series 32-bit microcontrollers are provided with 32 extension instructions that can be defined by users. By assigning extension functions that are suited for applications such as multiply, multiply-and-accumulate, saturation, and other operations for each model expansion and connecting the extension function unit through the extension function interface of the CPU core, it becomes possible to perform the desired processing faster.

Extension instructions include the instructions UDF00 through UDF15, which transfer the contents of a register or an immediate value through the extension function unit and then loads the operation results into a data register, and instructions UDF20 to UDF35, which only transfer the contents of a register to the extension function unit. Processing that performs user-defined operations and then immediately uses the result is assigned to the UDF00 through UDF15 instructions, while processing that only transfers data to the extension unit or that transfers data to the extension unit and then, after allowing several cycles for the execution of another instruction, fetches data from the extension unit is assigned to UDF20 to UDF35. Extension operations that require three or more inputs can be implemented by first using an instruction from UDF20 to UDF35 to transfer the input data to the extension function unit and then using an instruction from UDF00 to UDF15 to perform the operation.

A block diagram showing the extension function unit connected to the CPU core for this microcontroller series is shown below.

The MN103002A/MN103002AYB has a 32 × 32 multiplier, a 32 × 32 + 64 multiply-and-accumulate unit, a priority encoder, and a saturation compensation unit on chip. Extension instructions that use those extension function units are explained in the next section, "Extension Instructions."
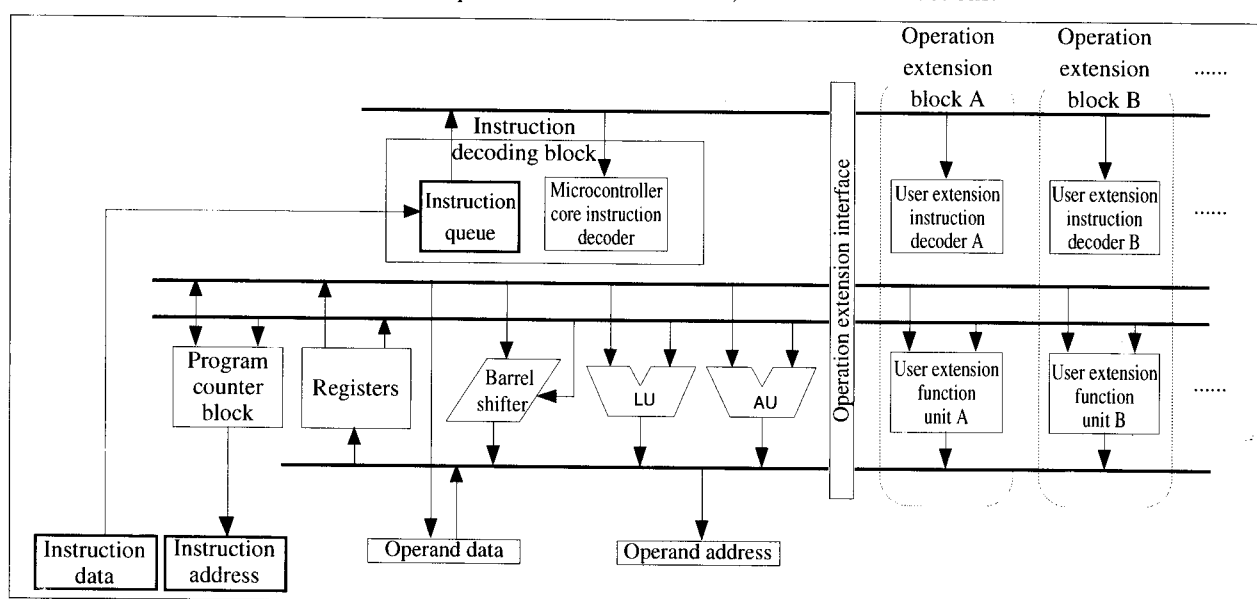
Fig. E-1 Block Diagram of the Extension Function Unit

Extension Instructions

Explanation of Notation

The notation that is used in the explanations of the extension instruction specifications is listed below:

OP:	Opcode
Am, An:	Address register (m, n = 3 to 0)
Dm, Dn:	Data register (m, n = 3 to 0)
SP:	Stack pointer
imm:	Immediate value (used in the general sense)
imm8:	8-bit immediate value
imm16:	16-bit immediate value
imm32:	32-bit immediate value
d8:	8-bit displacement
d16:	16-bit displacement
abs 16:	16-bit absolute
abs 32:	32-bit absolute
d32:	32-bit displacement
MDR:	Multiply/divide register (built into core)
MDRQ:	Quick multiply/divide register (built into Extension Function unit)
MCRL:	Multiply-and-accumulate register (built into Extension Function unit)
MCRH:	Multiply-and-accumulate register (built into Extension Function unit)
MCVF:	Multiply-and-accumulate flag (built into Extension Function unit)
LIR:	Loop instruction register
LAR:	Loop address register
PSW:	Program status word
PC:	Program counter
( ):	Indirect addressing
regs:	Multiple register specification
0x....:	Hexadecimal notation (The number that follows "0x" appears in hexadecimal notation.)

> **!** For details on indirect addressing, refer to section 2.5.2, "Addressing Modes."

The notation that is used to express flag changes is described below.

- : No change
+ : Flag change
* : Undefined
0 : Reset
1 : Set

"Flag" is the general term that is used to refer to the four lowest bits in the PSW (V, C, N, and Z).

Extension Function Unit Register Set

The extension function unit has the following dedicated registers in which it stores the results of quick multiplication operations and multiply-and-accumulate operations.

Multiply Register — 31   MDRQ   0

Multiply & Accumulate Register (Higher) — 31   MCRH   0

Multiply & Accumulate Register (Lower) — 31   MCRL   0

Multiply & Accumulate Overflow Detect Flag Register   MCVF   0
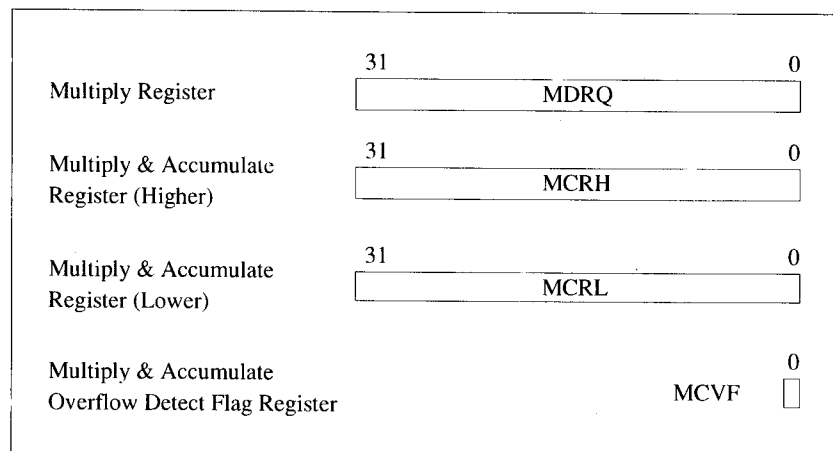
Fig. E-2 Extension Function Unit Register Set

Multiply register (32 bits × 1 register)

> This register is provided for high-speed multiplication instructions. A multiplication instruction uses this register to store the high-order 32 bits of the 64-bit multiplication result.

Multiply-and-accumulate register (higher) (32 bits × 1 register)

> This register is provided for multiply-and-accumulate operation instructions. A multiply-and-accumulate operation instruction uses this register to store the high-order 32 bits of the 64-bit multiply-and-accumulate operation result.

Multiply-and-accumulate register (lower) (32 bits × 1 register)

> This register is provided for multiply-and-accumulate operation instructions. A multiply-and-accumulate operation instruction uses this register to store the low-order 32 bits of the 64-bit multiply-and-accumulate operation result.

Multiply-and-accumulate overflow detect flag register (1 bit × 1 register)

> This one-bit register is set when an overflow occurs in a multiply-and-accumulate operation. This flag is not cleared until the next CLRMAC instruction or PUTCX instruction is executed.

Extension Instruction Details

PUTX (Register transfer instruction for quick multiplication)

[Instruction Format (Macro Name)]

       PUTX Dm

[Assembler Mnemonic]

       udf20 Dm,Dm

[Operation]

       This instruction transfers the contents of Dm to the quick multiply register MDRQ.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

       When "udf20 Dm,Dn" is operated, Dn is ignored.

       The operations of "udf20 imm8,Dn", "udf20 imm16,Dn" and "udf20 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

PUTCX (Multiply-and-accumulate register transfer instruction)

[Instruction Format (Macro Name)]

    PUTCX Dm,Dn

[Assembler Mnemonic]

    udf21 Dm,Dn

[Operation]

    This instruction transfers the contents of Dm to the multiply-and-accumulate register MCRH.

    This instruction also transfers the contents of Dn to the multiply-and-accumulate register MCRL.

    The contents of the V flag are set into the multiply-and-accumulate overflow detect register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

    The operations of "udf21 imm8,Dn", "udf21 imm16,Dn" and "udf21 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

GETX (Quick multiply register transfer instruction)

[Instruction Format (Macro Name)]

GETX Dn

[Assembler Mnemonic]

udf15 Dn,Dn

[Operation]

This instruction transfers the contents of the quick multiply register MDRQ to Dn.

[Flag Changes]

Flag	Change	Condition
V	0	Always "0"
C	0	Always "0"
N	+	"1" when the MSB of the transfer result is "1"; "0" in all other cases.
Z	+	"1" when the transfer result is "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the updating of the PSW to reflect flag changes.

When "udf15 Dm,Dn" is operated, Dm is ignored.

The operations of "udf15 imm8,Dn", "udf15 imm16,Dn" and "udf15 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

GETCHX (High-order 32 bit of multiply-and-accumulate register transfer instruction)

[Instruction Format (Macro Name)]

GETCHX Dn

[Assembler Mnemonic]

udf12 Dn,Dn

[Operation]

This instruction transfers the contents of the multiply-and-accumulate register MCRH to Dn.

The contents of the multiply-and-accumulate overflow detect register MCVF are set into the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

When "udf12 Dm,Dn" is operated, Dm is ignored.

The operations of "udf12 imm8,Dn", "udf12 imm16,Dn" and "udf12 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

GETCLX (Low-order 32 bit of multiply-and-accumlate register transfer instruction)

[Instruction Format (Macro Name)]

GETCLX Dn

[Assembler Mnemonic]

udf13 Dn,Dn

[Operation]

This instruction transfers the contents of the multiply-and-accumulate register MCRL to Dn.

The contents of the multiply-and-accumulate overflow detect register MCVF are set into the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

When "udf13 Dm,Dn" is operated, Dm is ignored.

The operations of "udf13 imm8,Dn", "udf13 imm16,Dn" and "udf13 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

CLRMAC (Multiply-and-accumulate register clean operation)

[Instruction Format (Macro Name)]

CLRMAC

[Assembler Mnemonic]

udf22 D0,D0

[Operation]

This instruction clears the contents of the multiply-and-accumulate registers MCRH and MCRL.

This instruction also clears the contents of the multiply-and-accumulate overflow detect register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

When "udf22 Dm,Dn" is operated, Dm and Dn are ignored.

The operations of "udf22 imm8,Dn", "udf22 imm16,Dn" and "udf22 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MULQ (Signed quick multiplication instruction: between registers)

[Instruction Format (Macro Name)]

MULQ Dm,Dn

[Assembler Mnemonic]

udf00 Dm,Dn

[Operation]

This instruction performs multiplication quickly using the multiplier in the extension function unit.

The instruction multiplies the contents of Dm (signed 32-bit integer: multiplicand) by the contents of Dn (signed 32-bit integer: multiplier), and then stores the upper 32 bits of the 64-bit result in the quick multiply register MDRQ and the lower 32 bits in Dn.

The range of significant values for the multiplicand that is stored in Dm before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the absolute value of the contents that are stored in Dm, the quicker the result of the operation can be derived.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the lower 32 bits of the result is "1"; "0" in all other cases.
Z	+	"1" when the lower 32 bits of the result are all "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

MULQI (Signed quick multiplication instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MULQI imm,Dn

[Assembler Mnemonic]

udf00	imm8,Dn	:	imm8 is sign-extended
udf00	imm16,Dn	:	imm16 is sign-extended
udf00	imm32,Dn		

[Operation]

This instruction performs multiplication quickly using the multiplier in the extension function unit.

The instruction multiplies the 32-bit data that is obtained by sign-extending imm (multiplicand) by the contents of Dn (signed 32-bit integer: multiplier), and then stores the upper 32 bits of the 64-bit result in the quick multiply register MDRQ and the lower 32 bits in Dn.

The range of significant values for the multiplicand that is stored in imm before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, if the number of imm bits is "16" or less, the operation results will be derived faster.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the lower 32 bits of the result is "1"; "0" in all other cases.
Z	+	"1" when the lower 32 bits of the result are all "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

MULQU (Unsigned quick multiplication instruction: between registers)

[Instruction Format (Macro Name)]

MULQU Dm,Dn

[Assembler Mnemonic]

udf01 Dm,Dn

[Operation]

This instruction performs multiplication quickly using the multiplier in the extension function unit.

The instruction multiplies the contents of Dm (unsigned 32-bit integer: multiplicand) by the contents of Dn (unsigned 32-bit integer: multiplier), and then stores the upper 32 bits of the 64-bit result in the quick multiply register MDRQ and the lower 32 bits in Dn.

The range of significant values for the multiplicand that is stored in Dm before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in Dm, the quicker the result of the operation can be derived.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the lower 32 bits of the result is "1"; "0" in all other cases.
Z	+	"1" when the lower 32 bits of the result are all "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

MULQIU (Unsigned quick multiplication instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MULQIU imm,Dn

[Assembler Mnemonic]

```
udfu01  imm8,Dn      :  imm8 is zero-extended
udfu01  imm16,Dn     :  imm16 is zero-extended
udfu01  imm32,Dn
```

[Operation]

This instruction performs multiplication quickly using the multiplier in the extension function unit.

The instruction multiplies the 32-bit data that is obtained by zero-extending imm (multiplicand) by the contents of Dn (unsigned 32-bit integer: multiplier), and then stores the upper 32 bits of the 64-bit result in the quick multiply register MDRQ and the lower 32 bits in Dn.

The range of significant values for the multiplicand that is stored in Dm before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, if the number of imm bits is "16" or less, the operation results will be derived faster.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the lower 32 bits of the result is "1"; "0" in all other cases.
Z	+	"1" when the lower 32 bits of the result are all "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

MAC (Signed multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MAC Dm,Dn

[Assembler Mnemonic]

udf28 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (signed 32-bit integer: multiplicand) by the contents of Dn (signed 32-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACI (Signed multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACI imm,Dn

[Assembler Mnemonic]

udf28	imm8,Dn	: imm8 is sign-extended
udf28	imm16,Dn	: imm16 is sign-extended
udf28	imm32,Dn	

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 32-bit data that is obtained by sign-extending imm (multiplicand) by the contents of Dn (signed 32-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACH (Signed half-word data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACH Dm,Dn

[Assembler Mnemonic]

udf30 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (signed 16-bit integer: multiplicand) by the contents of Dn (signed 16-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACIH (Signed half-word data multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACIH imm,Dn

[Assembler Mnemonic]

udf30 imm8,Dn   : imm8 is sign-extended

udf30 imm16,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 16-bit data that is obtained by sign-extending imm (multiplicand) by the contents of Dn (signed 16-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

The operations of "udf30 imm32,Dn" is not assured. In addition, a system error interrupt does not occur in these cases.

MACB (Signed byte data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACB Dm,Dn

[Assembler Mnemonic]

udf32 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (signed 8-bit integer: multiplicand) by the contents of Dn (signed 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.

If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACIB (Signed byte data multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACIB imm,Dn

[Assembler Mnemonic]

udf32 imm8,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 8-bit data imm (multiplicand) by the contents of Dn (signed 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.

If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

The operations of "udf32 imm16,Dn" and "udf32 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MACU (Unsigned multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACU Dm,Dn

[Assembler Mnemonic]

udf29 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (unsigned 32-bit integer: multiplicand) by the contents of Dn (unsigned 32-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACIU (Unsigned multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACIU imm,Dn

[Assembler Mnemonic]

udfu29 imm8,Dn   : imm8 is zero-extended
udfu29 imm16,Dn   : imm16 is zero-extended
udfu29 imm32,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 32-bit data that is obtained by zero-extending imm (multiplicand) by the contents of Dn (unsigned 32-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least two cycles must be inserted between this instruction and the next extension instruction.

MACHU (Unsigned half-word data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACHU Dm,Dn

[Assembler Mnemonic]

udf31 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (unsigned 16-bit integer: multiplicand) by the contents of Dn (unsigned 16-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACIHU (Unsigned half-word data multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACIHU imm,Dn

[Assembler Mnemonic]

udfu31 imm8,Dn   : imm8 is zero-extended

udfu31 imm16,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 16-bit data that is obtained by zero-extending imm (multiplicand) by the contents of Dn (unsigned 16-bit integer: multiplier), adds the upper 32 bits and lower 32 bits of the resulting 64-bit product to the upper and lower 32 bits, respectively, of the 64-bit cumulative sum that is stored in the multiply-and-accumulate registers MCRH and MCRL, and then stores the upper 32 bits of the new resulting cumulative sum back in multiply-and-accumulate register MCRH and the lower 32 bits in multiply-and-accumulate register MCRL.

If an overflow from the 64-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

The operations of "udfu31 imm32,Dn" is not assured. In addition, a system error interrupt does not occur in these cases.

MACBU (Unsigned byte data multiply-and-accumulate operation instruction: between registers)

[Instruction Format (Macro Name)]

MACBU Dm,Dn

[Assembler Mnemonic]

udf33 Dm,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the contents of Dm (unsigned 8-bit integer: multiplicand) by the contents of Dn (unsigned 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.

If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

MACIBU (Unsigned byte data multiply-and-accumulate operation instruction: between immediate value and register)

[Instruction Format (Macro Name)]

MACIBU imm,Dn

[Assembler Mnemonic]

udfu33 imm8,Dn

[Operation]

This instruction performs the multiply-and-accumulate operation using the multiplier and adder in the extension function unit.

The instruction multiplies the 8-bit data imm (multiplicand) by the contents of Dn (unsigned 8-bit integer: multiplier), adds the resulting product to the 32-bit cumulative sum that is stored in the multiply-and-accumulate register MCRL, and then stores the new resulting 32-bit cumulative sum back in multiply-and-accumulate register MCRL.

If an overflow from the 32-bit cumulative sum data is generated when the product is added to the cumulative sum, multiply-and-accumulate overflow detection flag 1 is output to register MCVF.

[Flag Changes]

Flag	Change	Condition
V	—	
C	—	
N	—	
Z	—	

[Programming Cautions]

A non-extension instruction that consumes at least one cycle must be inserted between this instruction and the next extension instruction.

The operations of "udfu33 imm16,Dn" and "udfu33 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SAT16(16-bit saturation operation instruction)

[Instruction Format (Macro Name)]

SAT16 Dm,Dn

[Assembler Mnemonic]

udf04 Dm,Dn

[Operation]

When Dm is a 16-bit signed number which is the maximum positive value (0x00007fff) or more, the maximum positive value (0x00007fff) is stored in Dn. When Dm is a 16-bit signed number which is the maximum negative value (0xffff8000) or less, the maximum negative value (0xffff8000) is stored in Dn. In all other cases, the contents of Dm are stored in Dn.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the operation result is "1"; "0" in all other cases.
Z	+	"1" when the operation result is "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf04 imm8,Dn", "udf04 imm16,Dn" and "udf04 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SAT24 (24-bit saturation operation instruction)

[Instruction Format (Macro Name)]

SAT24 Dm,Dn

[Assembler Mnemonic]

udf05 Dm,Dn

[Operation]

When Dm is a 24-bit signed number which is the maximum positive value (0x007fffff) or more, the maximum positive value (0x007fffff) is stored in Dn. When Dm is a 24-bit signed number which is the maximum negative value (0xff800000) or less, the maximum negative value (0xff800000) is stored in Dn. In all other cases, the contents of Dm are stored in Dn.

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	+	"1" when the MSB of the operation result is "1"; "0" in all other cases.
Z	+	"1" when the operation result is "0"; "0" in all other cases.

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf05 imm8,Dn", "udf05 imm16,Dn" and "udf05 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST (Multiply-and-accumulate operation results 8-, 16-, 32-bit saturation operation instruction)

[Instruction Format (Macro Name)]

MCST Dm,Dn

MCST imm8,Dn

[Assembler Mnemonic]

udf02 Dm,Dn

udf02 imm8,Dn   : Only 0x20, 0x10, and 0x08 are valid as values for imm8

[Operation]

This instruction sets the contents of the multiply-and-accumulate operation overflow detect register MCVF into the V flag. In addition, depending on the value of Dm or imm8, the following operations are performed.

(1) When the value of Dm or imm8 is 32 (0x00000020)

When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 32-bit signed number (0x000000007fffffff), the maximum positive value (0x7fffffff) is stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 32-bit signed number (0xffffffff80000000), the maximum negative value (0x80000000) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(2) When the value of Dm or imm8 is 16 (0x00000010)

When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 16-bit signed number (0x0000000000007fff), the maximum positive value (0x00007fff) is stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 16-bit signed number (0xffffffffffff8000), the maximum negative value (0xffff8000) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(3) When the value of Dm or imm8 is 8 (0x00000008)

When the 32-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate register MCRL is equal to or greater than the maximum positive value for an 8-bit signed number (0x0000007f), the maximum positive value (0x7f) is stored in Dn. If the value stored in the multiply-and-accumulate register MCRL is equal to or less than the maximum negative value for an 8-bit signed number (0xffffff80), the maximum negative value (0x80) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

(4) When the value of Dm or imm8 is any other value

The value in Dn is undefined.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf02 imm16,Dn" and "udf02 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST9 (Multiply-and-accumulate operation results 9-bit saturation operation instruction/positive value conversion instruction)

[Instruction Format (Macro Name)]

MCST9 Dn

[Assembler Mnemonic]

udf03 Dn,Dn

[Operation]

When the 32-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate register MCRL is equal to or greater than the maximum positive value for a 9-bit signed numeric value (0x000000ff), the maximum positive value (0xff) is stored in Dn. If the value stored in the multiply-and-accumulate register MCRL is equal to or less than the maximum negative value for a 32-bit signed numeric value (0x00000000), the maximum negative value (0x00) is stored in Dn. In all other cases, the contents of MCRL are stored in Dn.

This instruction also sets the contents of the multiply-and-accumulate operation overflow detect register MCVF in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

When "udf03 Dm,Dn" is operated, Dm is ignored.

The operations of "udf03 imm8,Dn", "udf03 imm16,Dn" and "udf03 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

MCST48 (Multiply-and-accumulate operation results 48-bit saturation operation instruction)

[Instruction Format (Macro Name)]

MCST48 Dn

[Assembler Mnemonic]

udf06 Dn,Dn

[Operation]

When the 64-bit result of the multiply-and-accumulate operation that is stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or greater than the maximum positive value for a 48-bit signed numeric value (0x00007fffffffffff), the maximum positive value (0x00007fffffffffff) is output and bits 47 through bits 16 of that output are stored in Dn. If the value stored in the multiply-and-accumulate registers MCRH and MCRL is equal to or less than the maximum negative value for a 48-bit signed numeric value (0xffff800000000000), the maximum negative value (0xffff800000000000) is output and bits 47 through bits 16 of that output are stored in Dn. In all other cases, the contents of MCRH and MCRL are output and bits 47 through bits 16 of that output are stored in Dn.

This instruction also sets the contents of the multiply-and-accumulate operation overflow detect register MCVF in the V flag.

[Flag Changes]

When multiply-and-accumulate operation overflow was not detected (MCVF = 0)

Flag	Change	Condition
V	0	Indicates that the multiply-and-accumulate operation is valid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

When multiply-and-accumulate operation overflow was detected (MCVF = 1)

Flag	Change	Condition
V	1	Indicates that the multiply-and-accumulate operation is invalid.
C	0	Always "0"
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

When "udf06 Dm,Dn" is operated, Dm is ignored.

The operations of "udf06 imm8,Dn", "udf06 imm16,Dn" and "udf06 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

BSCH (Bit Search)

[Instruction Format (Macro Name)]

BSCH Dm,Dn

[Assembler Mnemonic]

udf07 Dm,Dn

[Operation]

This instruction conducts a bit search within the 32-bit bit string stored in Dm, starting from the bit position of the bit number indicated by the contents of Dn - 1 and continuing in the direction of descending bit numbers. The bit number of the first bit position where a "1" is found is then stored in Dn.

When the least significant five bits of Dn are zeroes, the bit search is conducted from bit 31 and continues in the direction of descending bit numbers.

If the bit search reaches bit 0 without finding a "1", the "C" flag is set, 0x00000000 is written in Dn and execution of this instruction ends.

When execution of this instruction starts, the upper 27 bits of Dn are ignored.
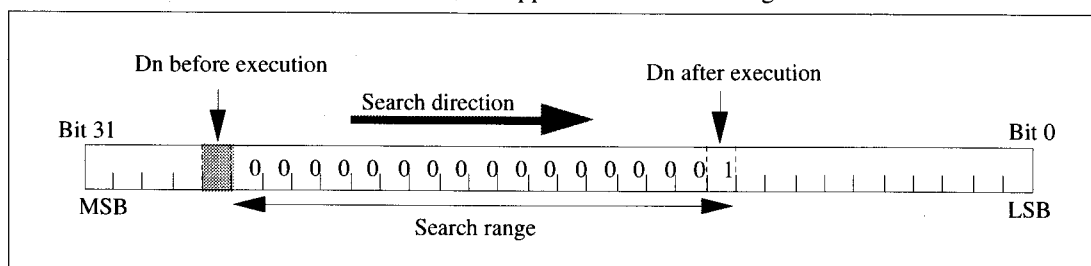
[Flag Changes]

When the search was successful (a "1" was found)

Flag	Change	Condition
V	*	Undefined
C	0	Indicates that the search was successful.
N	*	Undefined
Z	*	Undefined

When the search failed (no "1" was found)

Flag	Change	Condition
V	*	Undefined
C	1	Indicates that the search failed.
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf07 imm8,Dn", "udf07 imm16,Dn" and "udf07 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SWAP (Data swapping instruction that swaps bytes [high-order to low-order and vice versa] in four-byte data)

[Instruction Format (Macro Name)]

SWAP Dm,Dn

[Assembler Mnemonic]

udf08 Dm,Dn

[Operation]

This instruction swaps the positions of the high-order and low-order 8-bit bytes within the respective high- and low-order 16-bit half-words within the 32-bit data stored in Dm, and then swaps the positions of the high-order and low-order 16-bit half-words, and then stores the result in Dn. As a result, bits 31 through 24 of Dm are stored in bits 7 through 0 in Dn, bits 23 through 16 of Dm are stored in bits 15 through 8 in Dn, bits 15 through 8 of Dm are stored in bits 23 through 16 in Dn, and bits 7 through 0 of Dm are stored in bits 31 through 24 in Dn.
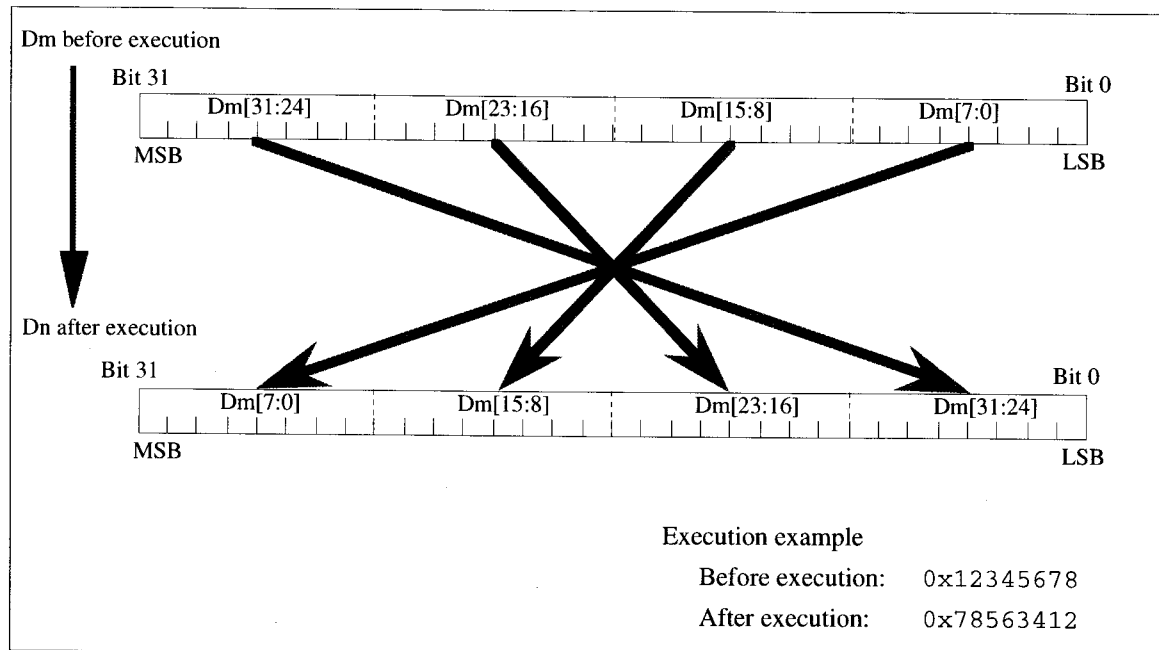
Execution example

Before execution: 0x12345678

After execution: 0x78563412

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf08 imm8,Dn", "udf08 imm16,Dn" and "udf08 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

SWAPH (Data swapping instruction [high-order to low-order and vice versa] in two-byte data)

[Instruction Format (Macro Name)]

SWAPH Dm,Dn

[Assembler Mnemonic]

udf09 Dm,Dn

[Operation]

This instruction swaps bits 15 through 8 of Dm with bits 7 through 0, and bits 31 through 24 with bits 23 through 16, and then stores the result in Dn.
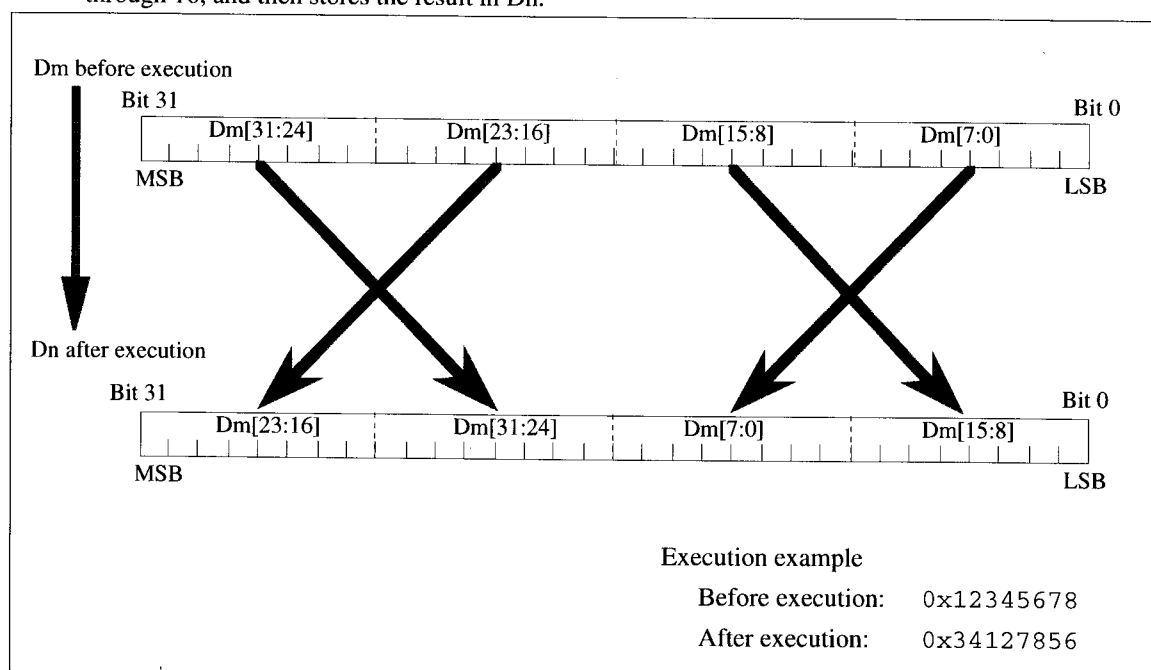
Execution example

Before execution: 0x12345678

After execution: 0x34127856

[Flag Changes]

Flag	Change	Condition
V	*	Undefined
C	*	Undefined
N	*	Undefined
Z	*	Undefined

[Programming Cautions]

There is a one-instruction delay in the updating of the PSW to reflect flag changes.

However, the Bcc and Lcc instructions can evaluate the flags without waiting for the flag changes to be reflected in the PSW.

The operations of "udf09 imm8,Dn", "udf09 imm16,Dn" and "udf09 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

Notes on Extension Operation Programming

The extension function unit is equipped with the following dedicated registers that it uses to store the results of quick multiplication operations and multiply-and-accumulate operations.

(1) Notes on instruction description

These programming notes address instruction descriptions as well as instruction placement and combinations. Failure to heed these notes will result in misoperation. A list of these notes is shown below.

Table E-1 List of Notes

Preceding instruction	Following instruction	Placement relationship	Notes
Word/half-word data Multiply-and-accumulate instruction *1	Quick multiplication instruction *3	Following	Insert at least one cycle between the instructions
Word/half-word data Multiply-and-accumulate instruction *1	Multiply-and-accumulate instruction *4	Following	Insert at least two cycles between the instructions
Word/half-word data Multiply-and-accumulate instruction *1	MCRH, MCRL access instruction *5	Following	Insert at least three cycles between the instructions
Byte data Multiply-and-accumulate instruction *2	Multiply-and-accumulate operation instruction *4	Following	Insert at least one cycle between the instructions
Byte data Multiply-and-accumulate instruction *2	MCRH, MCRL access instruction *5	Following	Insert at least two cycles between the instructions
mov [regs], SP instruction	Extension Operation instruction	Following	Insert at least one cycle between the instructions *6

*1: The category "Word/half-word data multiply-and-accumulate instructions" applies to the following instructions: MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction

*2: The category "byte data multiply-and-accumulate instructions" applies to the following instructions: MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

*3: The category "Quick multiplication instructions" applies to the following instructions: MULQ instruction, MULQI instruction, MULQU instruction, MULQIU instruction

*4: The category "Multiply-and-accumulate operation instructions" applies to the following instructions: MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction, MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

*5: The category "MCRH, MCRL access instructions" applies to the following instructions: PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

*6: This problem can be avoided by an assembler after V3.1R9. At assembling, nop instruction is automatically inserted between the mov and extended operation instructions.

(1-a) Note on the description of word/half-word data multiply-and-accumulate instructions and quick multiplication instructions

A common function unit is used to execute word/half-word data multiply-and-accumulate instructions and quick multiplication instructions. Therefore, when executing a word/half-word data multiply-and-accumulate instruction on the common function unit, it is essential to not initiate the subsequent quick multiplication instruction until after execution of the word/half-word data multiply-and-accumulate instruction has been completed. In addition, one cycle must be inserted between the word/half-word data multiply-and-accumulate instruction and the subsequent quick multiplication instruction.
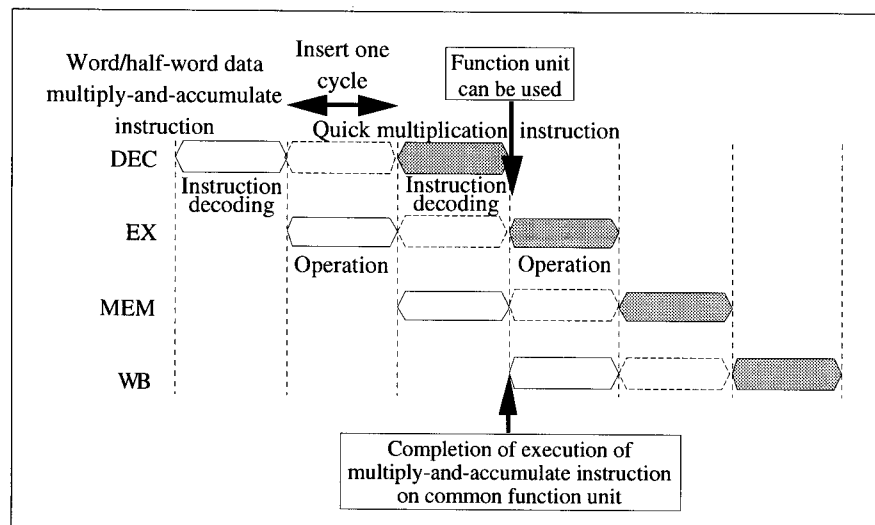
Fig. E-3 Pipeline Diagram Illustrating This Note (1)

This note applies to the following instructions:

    <Word/half-word data multiply-and-accumulate instructions>

    MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction

    <Quick multiplication instruction>

    MULQ instruction, MULQI instruction, MULQU instruction, MULQIU instruction

(1-b) Note on the description of word/half-word data multiply-and-accumulate instructions and multiply-and-accumulate instructions

When executing a word/half-word data multiply-and-accumulate instruction followed by a multiply-and-accumulate instruction, the result produced by the word/half-word data multiply-and-accumulate instruction is used in the execution of the subsequent multiply-and-accumulate instruction. Therefore, it is essential to not initiate the subsequent multiply-and-accumulate instruction until after the required result of the word/halfword data multiply-and-accumlate instruction has been output. In addition, two cycles must be inserted between the word/half-word data multiply-and-accumulate instruction and the subsequent multiply-and-accumulate instruction.

Fig. E-4 Pipeline Diagram Illustrating This Note (2)

This note applies to the following instructions:

<Word/half-word data multiply-and-accumulate instructions>

MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction

<Multiply-and-accumulate instructions>

MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction, MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

(1-c) Note on the description of word/half-word data multiply-and-accumulate instructions and MCRH, MCRL access instructions

When executing a word/half-word data multiply-and-accumulate instruction followed by an MCRH, MCRL access instruction, the result produced by the word/half-word data multiply-and-accumulate instruction is used in the execution of the subsequent MCRH, MCRL access instruction. Therefore, it is essential to not initiate the subsequent MCRH, MCRL access instruction until after the required result of the word/halfword data multiply-and-accumlate instruction has been output. In addition, three cycles must be inserted between the word/half-word data multiply-and-accumulate instruction and the subsequent MCRH, MCRL access instruction.
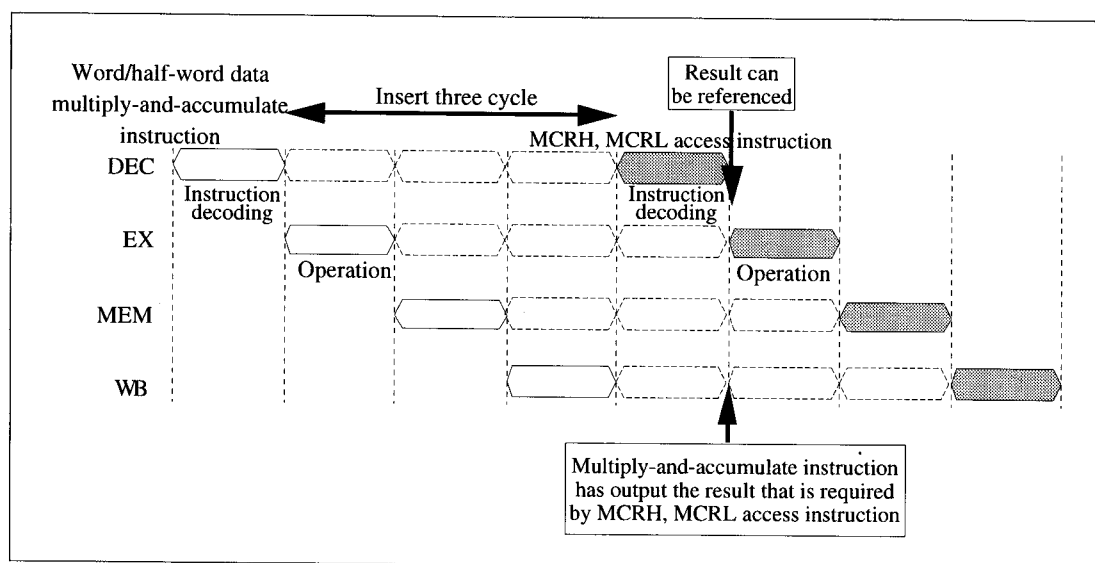
Fig. E-5 Pipeline Diagram Illustrating This Note (3)

This note applies to the following instructions:

    <Word/half-word data multiply-and-accumulate instructions>

    MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction

    <MCRH, MCRL access instructions>

    PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

(1-d) Note on the description of byte data multiply-and-accumulate instructions and multiply-and-accumulate instructions

When executing a byte data multiply-and-accumulate instruction followed by a multiply-and-accumulate instruction, the result produced by the byte data multiply-and-accumulate instruction is used in the execution of the subsequent multiply-and-accumulate instruction. Therefore, it is essential to not initiate the subsequent multiply-and-accumulate instruction until after the required result of the byte data multiply-and-accumlate instruction has been output. In addition, one cycles must be inserted between the byte data multiply-and-accumulate instruction and the subsequent multiply-and-accumulate instruction.
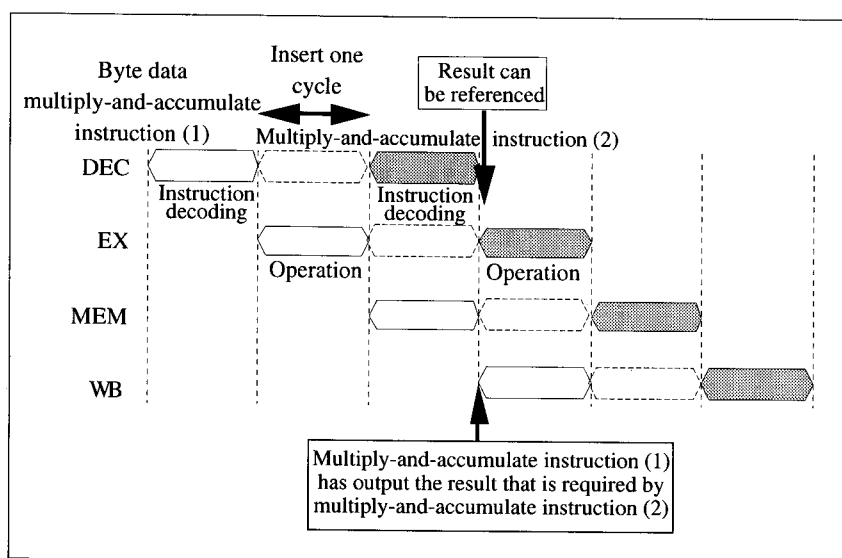
Fig. E-6 Pipeline Diagram Illustrating This Note (4)

This note applies to the following instructions:

<Byte data multiply-and-accumulate instructions>

MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

<Multiply-and-accumulate instructions>

MAC instruction, MACI instruction, MACH instruction, MACIH instruction, MACU instruction, MACIU instruction, MACHU instruction, MACIHU instruction, MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

(1-e) Note on the description of byte data multiply-and-accumulate instructions and MCRH, MCRL access instructions

When executing a byte data multiply-and-accumulate instruction followed by an MCRH, MCRL access instruction, the result produced by the byte data multiply-and-accumulate instruction is used in the execution of the subsequent MCRH, MCRL access instruction. Therefore, it is essential to not initiate the subsequent MCRH, MCRL access instruction until after the required result of the byte data multiply-and-accumlate instruction has been output. In addition, two cycles must be inserted between the byte data multiply-and-accumulate instruction and the subsequent MCRH, MCRL access instruction.

Fig. E-7 Pipeline Diagram Illustrating This Note (5)

This note applies to the following instructions:

      <Byte data multiply-and-accumulate instructions>

      MACB instruction, MACIB instruction, MACBU instruction, MACIBU instruction

      <MCRH, MCRL access instructions>

      PUTCX instruction, CLRMAC instruction, GETCHX instruction, GETCLX instruction

Extension Instruction List (Includes Code Length, Number of Cycles)

Instruction		Source	Destination	Format	Code Length	Number of Cycles	Remarks
PUTX	PUTX	Dm		D0	2	1	
	PUTCX	Dm	Dn	D0	2	1	
GETX	GETX		Dn	D0	2	1	
	GETCHX		Dn	D0	2	1	
	GETCLX		Dn	D0	2	1	
CLRMAC	CLRMAC			D0	2	1	
MULQ	MULQ	Dm	Dn	D0	2	4	Dm is a value which can be expressed with 2 bytes to 1 byte or Dm = 0.
						5	Dm is a value which can be expressed with 4 bytes to 3 bytes.
	MULQI	imm8	Dn	D1	3	4	
	MULQI	imm16	Dn	D2	4	4	
	MULQI	imm32	Dn	D4	6	5	imm32 is a value which can be expressed with 2 bytes to 1 byte or imm32 = 0.
						6	imm32 is a value which can be expressed with 4 bytes to 3 bytes.
MULQU	MULQU	Dm	Dn	D0	2	4	Dm is a value which can be expressed with 2 bytes to 1 byte or Dm = 0.
						5	Dm is a value which can be expressed with 4 bytes to 3 bytes.
	MULQIU	imm8	Dn	D1	3	4	
	MULQIU	imm16	Dn	D2	4	4	
	MULQIU	imm32	Dn	D4	6	5	imm32 is a value which can be expressed with 2 bytes to 1 byte or imm32 = 0.
						6	imm32 is a value which can be expressed with 4 bytes to 3 bytes.
MAC	MAC	Dm	Dn	D0	2	1	
	MACI	imm8	Dn	D1	3	1	
	MACI	imm16	Dn	D2	4	1	
	MACI	imm32	Dn	D4	6	2	
	MACH	Dm	Dn	D0	2	1	
	MACIH	imm8	Dn	D1	3	1	
	MACIH	imm16	Dn	D2	4	1	
	MACB	Dm	Dn	D0	2	1	
	MACIB	imm8	Dn	D1	3	1	
MACU	MACU	Dm	Dn	D0	2	1	
	MACIU	imm8	Dn	D1	3	1	
	MACIU	imm16	Dn	D2	4	1	
	MACIU	imm32	Dn	D4	6	2	
	MACHU	Dm	Dn	D0	2	1	
	MACIHU	imm8	Dn	D1	3	1	
	MACIHU	imm16	Dn	D2	4	1	
	MACBU	Dm	Dn	D0	2	1	
	MACIBU	imm8	Dn	D1	3	1	
SAT16	SAT16	Dm	Dn	D0	2	2	
SAT24	SAT24	Dm	Dn	D0	2	2	
MCST	MCST	Dm	Dn	D0	2	2	
	MCST	imm8	Dn	D0	2	2	
	MCST9		Dn	D0	2	2	
	MCST48		Dn	D0	2	2	
BSCH	BSCH	Dm	Dn	D0	2	2	
SWAP	SWAP	Dm	Dn	D0	2	1	
	SWAPH	Dm	Dn	D0	2	1	

Appendix F: Package Outline and Dimensions

Fig.F-1 shows the package outline and dimensions of the MN103002A, and Fig.F-2 shows the package outline and dimensions of the MN103002AYB.

Package code: *QFP160-P-2828B

Unit: mm

Sealing material: Epoxy resin

Leads: Copper alloy

Lead surface treatment: Solder plating

Fig. F-1 Package Outline and Dimensions of MN103002A

单位 : mm

Package code: FLGA165-C-1111

11.00±0.15

11.00±0.15

TOP VIEW

0.85±0.15
0.05max
0.05

BOTTOM VIEW

0.70±0.20
0.70±0.20
9.60
9.60
Ø0.50±0.10
Ø0.15 M
0.80

N M L K J H G F E D C B A

1 2 3 4 5 6 7 8 9 10 11 12 13

Sealing resin: Epoxy resin
Body material: Ceramic
Lead finish method: Au plating

Fig. F-2 Package Outline and Dimensions of MN103002AYB

The correction table in The Revised Edition of MN103002A LSI User's Manual
(From 3rd Edition 1st Printing to 5th Edition)

Page	Errors	Page	Corrections
-		-	(MN103002AYB (CSP) has been added to the products described in this LSI User's Manual since 5th edition. Accordingly, the product name is changed partly in writing.)
P.2	(In the 7th line of "1.1 Overview.") ... 3.3 V ± 5%, ...	P.2	(In the 7th line of "1.1 Overview.") ... 3.3 V ± 0.165 V, ...
P.2		P.2	(The following sentence is added at the end of "1.1 Overview.") MN103002AYB is a product with MN103002A package changed to CSP, and includes the same function and performance as MN103002A.
P.3	• Supports high-speed page mode.	P.3	• Supports high-speed page mode. (Supports the page mode mix cycle DRAM.)
P.3	● The DRAM interface can be used with an external bus master (block 3 only)	P.3	(Deleted.)
P.4	● Interrupts • 30 sources - External interrupts: 9 sources (8 individual IRQs, and 1 external NMI)	P.4	● Interrupts • 30 sources - External interrupts: 9 sources ($\overline{\text{IRQn}}$ (n=7 to 0) x 8, and $\overline{\text{NMIRQ}}$ x 1)
P.4		P.4	(Following item is added.) ● Package • MN103002A : QFP160-P-2828B (28 mm square, 0.65 mm pitch QFP) • MN103002AYB: FLGA165-C-1111 (11 mm square, 0.8 mm pitch CSP)
-		P.8	(Figure 1-4-2 MN103002AYB Pin Assignment Diagram is added. As the result of adding the diagram and the change on p.9, the pin function list moved to p.10-p.11, and became Table 1-4-3.)
-		P.9	(Table 1-4-2 MN103002AYB Pin Assignments is added. As the result of adding the pin assignments and the change on p.8, the pin function list moved to p.10-p.11, and became Table 1-4-3.)
P.8		P.10	(Table title is added to a table of section 1.4.2.)
P.9	(The column of "Pin Function" such as "Pin name" is "$\overline{\text{NMIRQ}}$" in the table.) External NMI signal input	P.11	(The column of "Pin Function" such as "Pin name" is "$\overline{\text{NMIRQ}}$" in the table.) External non-maskable interrupt signal input
P.12	● Minimum instruction execution cycle: 1 cycle (15 nsec)	P.14	● Maximum instruction execution cycle: 1 cycle (15 ns)
P.17	(Omit)	P.19	(CPU pipeline control register was deleted from the table 2-3-1, List of Control Registers.)
P.19	(Omit)	-	(CPU Pipeline Control Register was deleted.)
P.22		P.23	(Figure title is added to a figure of section 2.5.1.)
P.24		P.25	
P.25	(In the Sign extension column of the transfer instructions) EXT EXTB EXTBU EXTH EXT	P.26	(In the Sign extension column of the transfer instructions) EXT EXTB EXTBU EXTH EXTHU
P.25	Note: Interrupts are prohibited and the bus is locked (occupied by the CPU) when BSET or BCLR is being executed.	P.26	(A proviso is added to the left-mentioned Note.)
P.32	(From 8th line to 9th line of "Explanation of the Memory Map.") Note that operation is not guaranteed when accessing an unmounted space, such as when accessing an internal I/O space to which no control register has been allocated.	P.33	(From 8th line to 9th line of "Explanation of the Memory Map.") Note that the unmounted space access is prohibited such as access to the internal I/O space without a control register. The operation is not assured.
P.33	Reset interrupt Highest priority ranking NMI interrupts : Level interrupt n (n = 0 to 6) Lowest priority ranking	P.34	Reset interrupt Highest priority ranking Non-maskable interrupts : Level interrupt n (n = 0 to 6) Lowest priority ranking
P.35		P.36	
P.36	(In the 5th line of "2.7.2.2 Interrupt Control Registers (GnICR)") G0ICR is an NMI-only register, ...	P.37	(In the 5th line of "2.7.2.2 Interrupt Control Registers (GnICR)") G0ICR is dedicated for non-maskable interrupts, ...
P.37		P.38	(Table number is added to a table of section 2.7.2.2.)
P.38	(In the 9th line of "2.7.2.3 Interrupt Accept Group Registers (IAGR)") Accessing the IAGR has no meaning during an NMI interrupt.	P.39	(In the 9th line of "2.7.2.3 Interrupt Accept Group Registers (IAGR)") Accessing the IAGR has no meaning during non-maskable interrupts.

Page	Errors	Page	Corrections
P.38	(In the 2nd line of "**External pin non-maskable interrupts**") ..., the external NMI request flag (NMIF) in the ...	P.40	(In the 2nd line of "**External pin non-maskable interrupts**") ..., the external non-maskable interrupt request flag (NMIF) in the ...
P.39	(The 1st line to 2nd line of "**System error interrupts**") A system error interrupt is generated when a fatal error occurs, such as when an attempt is made to execute an unimplemented instruction.	P.40	(The 1st line to 2nd line of "**System error interrupts**") System error interrupt occurs when an unaligned memory access or an unimplemented instruction is executed or other fatal error occurs.
P.39	**An Interrupt Status Register...**	P.40	(Deleted)
P.39		P.40	(Following note is added.) **Do not change the interrupt enable (IE) in PSW during non-maskable interrupt proccessing.**
P.40	(In "**Hardware interrupt processing sequence**") Step.1　The PC (the return address) is saved to the stack (SP-4). Step.2　The contents of the PSW are saved to the stack. (SP-8). Step.3　The contents of the PSW are updated. 　　　　IE is cleared and .... (In the case of an NMI, IM2 to IM0 are undefined.)	P.42	(In "**Hardware interrupt processing sequence**") Step.1　The contents of the PSW are saved to the stack. (SP-8). Step.2　The PC (the return address) is saved to the stack (SP-4). Step.3　The contents of the PSW are updated. 　　　　IE is cleared and .... (In the case of non-maskable interrupts, IM2 to IM0 are undefined.)
P.41	(In the number 3 of "**Example of preprocessing by the interrupt handler**") • In the case of an NMI, ...	P.43	(In the number 3 of "**Example of preprocessing by the interrupt handler**") • In the case of non-maskable interrupts, ...
P.45	(In Fig. 2-8-1) HALT ← Interrupt // //	P.47	(In Fig. 2-8-1) HALT ← Interrupt // Program //
P.46	(In the mathematical expression.)　15.888 msec	P.48	(In the mathematical expression.)　15.888 ms
P.56	(The 1st line of "4.2 Features") The features of the MN103002A's built-in caches are...	P.58	(The 1st line of "4.2 Features") The features of the built-in caches of the MN103002A/MN103002AYB are ...
P.58	Fig. 4-1	P.60	Fig. 4-3-1
P.59	Fig. 4-2	P.61	Fig. 4-3-2
P.59	(In Fig. 4-2) ← 22 bit →	P.61	(In Fig. 4-3-2) ← 23 bits →
P.60	("Purpose" of Cache Control Register.)	P.62	("Purpose" of Cache Control Register.) This register sets the operaton of caches.
P.63	Fig. 4-3	P.65	Fig. 4-5-1
P.64	Fig. 4-4	P.66	Fig. 4-5-2
P.65	Fig. 4-5	P.67	Fig. 4-5-3
P.67	Fig. 4-6	P.69	Fig. 4-5-4
P.68	Fig. 4-7	P.70	Fig. 4-5-5
P.68	(Following description in the figure.) CHCTR [9:8]	P.70	(Following description in the figure.) CHCTR [13:12]
P.69	Fig. 4-8	P.71	Fig. 4-5-6
P.70	Fig. 4-9	P.72	Fig. 4-5-7
P.71	Fig. 4-7	P.73	Fig. 4-5-5
P.71	Fig. 4-10	P.73	Fig. 4-5-8
P.73	Fig. 4-11	P.75	Fig. 4-5-9
P.74	Fig. 4-12	P.76	Fig. 4-5-10
P.75	Fig. 4-13	P.77	Fig. 4-5-11
P.77	Fig. 4-14	P.79	Fig. 4-6-1

Page	Errors	Page	Corrections
P.78	Fig. 4-15	P.80	Fig. 4-6-2
P.80	- Support for high-speed page mode.	P.82	- Supports high-speed page mode. (Supports the page mode mix cycle DRAM.)

P.88 (Description of Bit No. 12 to 8)

The relationship between the value of B0WC4 to 0 and the number of wait states is shown below.

CPU clock frequency MCLK (MHz)	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)	
		When FRQ = 0	When FRQ = 1
26.0 ⩽ MCLK ⩽ 52.0	x0000:Setting prohibited x0001:1 wait states x0010:2 wait states x0011:3 wait states : x1111:15 wait states	00000:Setting prohibited 00001:1 wait states 00010:2 wait states 00011:3 wait states : 11111:31 wait states	00000:Setting prohibited 00001:1 wait states 00010:2 wait states 00011:3 wait states : 11111:31 wait states
52.0 ⩽ MCLK ⩽ 66.7	x0000:Setting prohibited x0001:1 wait states x0010:2 wait states x0011:3 wait states : x1111:15 wait states		

P.90 to P.91 (Description of Bit No. 12 to 8)

The relationship between the value of B0WC4 to 0 and the number of wait states is shown below.

(1) When not using the in-circuit emulator (PX-ICE103002)

Basic bus cycle	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQ = 0	x0000：Setting prohibited x0001：1 wait states x0010：2 wait states x0011：3 wait states : x1111：15 wait states	00000：Setting prohibited 00001：1 wait states 00010：2 wait states 00011：3 wait states : 11111：31 wait states
FRQ = 1		00000：Setting prohibited 00001：Setting prohibited 00010：Setting prohibited 00011：3 wait states : 11111：31 wait states

(2) When using the in-circuit emulator (PX-ICE103002)

[1] When using the emulation function

External input pin	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQS = L	x0000：Setting prohibited x0001：Setting prohibited x0010：2 wait states x0011：3 wait states : x1111：15 wait states	00000：Setting prohibited 00011：Setting prohibited 00100：4 wait states : 11111：31 wait states
FRQS = H	x0000：Setting prohibited x0001：1 wait states x0010：2 wait states x0011：3 wait states : x1111：15 wait states	

[2] When using the external trace function

External input pin	Synchronous mode [B0BM = 0] (number of wait states counted by SYSCLK)	Asynchronous mode [B0BM = 1] (number of wait states counted by MCLK)
FRQS = L	x0000：Setting prohibited x0001：1 wait states x0010：2 wait states x0011：3 wait states : x1111：15 wait states	00000：Setting prohibited 00001：Setting prohibited 00010：Setting prohibited 00011：3 wait states : 11111：31 wait states
FRQS = H		

P.98 (Description of Bit No. 11 to 8)

The relationship between the value of WC3 to 0 and the number of wait states is shown below.

When FRQ = 0 (number of wait states counted by MCLK)	When FRQ = 1 (number of wait states counted by MCLK)
0000：0 wait states 0001：1 wait states 0010：2 wait states 0011：3 wait states : 0111：7 wait states	0000：0 wait states 0010：2 wait states 0100：4 wait states 0110：6 wait states : 1110：14 wait states (Setting an odd number of wait states is prohibited.)

DRAM mode is not valid when DRAME is "0" and BnDRAM in the MEMCTRn register is "1".

The refresh operation is not performed when DRAME is "0" and REFE is "1".

P.98 to P.100 (Description of Bit No. 11 to 8)

The relationship between the value of WC3 to 0 and the number of wait states is shown below.

(1) When not using the in-circuit emulator (PX-ICE103002)

When FRQ = 0 (number of wait states counted by MCLK)	When FRQ = 1 (number of wait states counted by MCLK)
0000：0 wait states 0001：1 wait states 0010：2 wait states 0011：3 wait states : 0111：7 wait states	0000：0 wait states 0010：2 wait states 0100：4 wait states 0110：6 wait states : 1110：14 wait states (Setting an odd number of wait states is prohibited.)

<Continued>

Page	Errors	Page	Corrections
P.98		P.98 to P.100	<Continued> (2) When using the in-circuit emulator (PX-ICE103002) [1] When using the emulation function

<table>
<tr><td>Page mode</td><td>When FRQ = 0
(number of wait states
counted by MCLK)</td><td>When FRQ = 1
(number of wait states
counted by MCLK)</td></tr>
<tr><td>Used
PAGE = 0</td><td>0000：Setting prohibited
0001：Setting prohibited
0010：2 wait states
0011：3 wait states
：
0111：7 wait states</td><td>0000：0 wait states
0010：2 wait states
0100：4 wait states
0110：6 wait states
：
1110：14 wait states
(Setting an odd number of
wait states is prohibited.)</td></tr>
<tr><td>Not used
PAGE = 1</td><td>0000：Setting prohibited
0001：Setting prohibited
0010：Setting prohibited
0011：3 wait states
：
0111：7 wait states</td><td>0000：Setting prohibited
0010：2 wait states
0100：4 wait states
0110：6 wait states
：
1110：14 wait states
(Setting an odd number of
wait states is prohibited.)</td></tr>
</table>

[2] When using the external bus trace function

<table>
<tr><td>Page mode</td><td>When FRQ = 0
(number of wait states
counted by MCLK)</td><td>When FRQ = 1
(number of wait states
counted by MCLK)</td></tr>
<tr><td>Used
PAGE = 0</td><td>0000：0 wait states
0001：1 wait states
0010：2 wait states
0011：3 wait states
：
0111：7 wait states</td><td rowspan="2">0000：0 wait states
0010：2 wait states
0100：4 wait states
0110：6 wait states
：
1110：14 wait states
(Setting an odd number of
wait states is prohibited.)</td></tr>
<tr><td>Not used
PAGE = 1</td><td>0000：Setting prohibited
0001：Setting prohibited
0010：2 wait states
0011：3 wait states
：
0111：7 wait states</td></tr>
</table>

> **When BnDRAM of MEMCTRn is "1"and DRAME is "0", DRAM mode is not valid.**
> **The refresh operation is not performed when REFE is "1" and DRAME is "0".**

Page	Errors	Page	Corrections
P.102	(In table 5-8-1) FRQS = L MCLK　　40 MHz SYSCLK　20 MHz IOCLK　　10 MHz	P.104	(In table 5-8-1) FRQS = L MCLK　　66 MHz SYSCLK　33 MHz IOCLK　16.5 MHz
P.107	Fig. 5-8-4 Timing for Access in Synchronous Mode ...	P.109	Fig. 5-8-4 Timing for Access in 32-bit Bus Synchronous Mode ...
P.107	Fig. 5-8-5 Timing for Access in Synchronous Mode ...	P.109	Fig. 5-8-5 Timing for Access in 32-bit Bus Synchronous Mode ...
P.108	Fig. 5-8-6 Timing for Bus Access by Handshaking	P.110	Fig. 5-8-6 Timing for Bus Access by 32-bit Bus Handshaking
P.109	Fig. 5-8-7 Timing for Access in Synchronous Mode ...	P.111	Fig. 5-8-7 Timing for Access in 16-bit Bus Synchronous Mode ...
P.110	Fig. 5-8-8 Timing for Access in Synchronous Mode ...	P.112	Fig. 5-8-8 Timing for Access in 16-bit Bus Synchronous Mode ...
P.110	Fig. 5-8-9 Timing for Bus Access by Handshaking	P.112	Fig. 5-8-9 Timing for Bus Access by 16-bit Bus Handshaking
P.111	Fig. 5-8-10 Timing for Access in Asynchronous Mode ...	P.113	Fig. 5-8-10 Timing for Access in Asynchronous Mode with 32-bit Bus ...
P.112	Fig. 5-8-11 Timing for Access in Asynchronous Mode ...	P.114	Fig. 5-8-11 Timing for Access in Asynchronous Mode with 32-bit Bus ...
-		P.134	("5.9 Cautions" is added.)
P.137	(The Address of DM0CTR)　　　x'3200100	P.139	(The Address of DM0CTR)　　　x'32000100
P.146	(In section 6.3.4) H'0000：1 transfer X'FFFF：65,536 transfers	P.148	(In sectoin 6.3.4) x'0000：1 transfer x'FFFF：65536 transfers
-		P.154	("Notes on the execution of a BSET or BCLR instruction." is added.)
P.156	(Clock name in Fig. 6-4-8)　MCLK	P.159	(Clock name in Fig. 6-4-8)　SYSCLK
P.156	(In Fig. 6-4-8)　$\overline{\text{DMA}}$ sampling	P.159	(In Fig. 6-4-8)　$\overline{\text{DMR}}$ sampling

Page	Errors	Page	Corrections
P.162	(In Fig. 7-3-2)	P.164	(In Fig. 7-3-2)
P.167	("Purpose" of Non-maskable Interrupt Control Register.) Governs the generation of NMIs.	P.169	("Purpose" of Non-maskable Interrupt Control Register.) Governs the generation of a non-maskable interrupt.
P.167	Bit name Description NMIF External NMI request flag	P.169	Bit name Description NMIF External non-maskable interrupt request flag

P.167	(From 1st line of main text.)
	When an non-maskable interrupt request is generated, the corresponding flag is set.
	After the non-maskable interrupt is accepted, the corresponding flag is cleared by software in the non-maskable interrupt handler. When a flag is set to "1", the flag is cleared by writing a "1" to it. The relationship between the flag status, the write data, and the value of the flag after the write is indicated in the table below.
	<The table of flag status is here.>
	
	! NMIs cannot be generated by software.
		P.169	(From 1st line of main text.)
			The method of clearing flag differs according to the interrupt request flags.
			1. External non-maskable interrupt request flag (NMIF) and Watchdog timer overflow interrupt request flag (WDIF)
			  After a non-maskable interrupt is accepted, these flags can be cleared by writing to the non-maskable interrupt control register (NMICR). When a flag is set to "1", write a "1" to the flag to clear it. The relationship between the flag status, the data written to the flag, and the new flag status after the data is written is shown in the table below.
			    <The table of flag status is here.>
			2. System error interrupt request flag (SYSEF)
			  This flag cannot be cleared by writing to the non-maskable interrupt control register (NMICR).
			  This flag can be cleared by generating a reset interrupt by setting the $\overline{RST}$ pin to "L" level or by the self-reset, which is generated by writing to the reset control register (RSTCTR) of the watchdog timer.
			
			! Non-maskable interrupts cannot be generated by software.

P.200	(The 2nd line right after the itemizations.) ... it is a non-maskable interrupt (NMI) or a level interrupt.	P.202	(The 2nd line right after the itemizations.) ... it is a non-maskable interrupt or a level interrupt.
P.200	(The 2nd line from the bottom.) ..., an NMI interrupt request is sent to the CPU ...	P.202	(The 2nd line from the bottom.) ..., a non-maskable interrupt request is sent to the CPU ...
P.205	Fig. 8-3-1 Timer Configuration Diagram	P.207	Fig. 8-3-1 Timer Connection Diagram
P.228	(The 1st line of main text) Prescaler control register (TMPSCNT) is used at 8 bit controller.	P.230	(The 1st line of main text) Prescaler control register (TMPSCNT) is also used at 8-bit timer.
P.239	9.7.1.1 Register Settings	P.241	9.7.1.1 Interval Timer, Timer Output
P.264	(In "● Oscillation stabilization wait time") When recovering from STOP mode: 3.972 ms to 1016.801 ms	P.266	(In "● Oscillation stabilization wait time") When recovering from STOP mode: 15.888 ms
P.270	(From 4th line to 6th line.) When recovering from STOP mode, ... is 16.5 MHz)	P.272	(Deleted.)
P.271	(In fig.10-5-2) 3.972 to 1016.801 ms	P.273	(In fig.10-5-2) 15.888 ms
P.307		P.309	(Figure title is added to a figure of section 11.5.2.1.)
P.332	(The address of "Port 4 Dedicated Output Control Register") Register symbol: P4SS Address: x'36008084	P.334	(The address of "Port 4 Dedicated Output Control Register") Register symbol: P4SS Address: x'36008048
P.337 to P.364	$T_A = -20 \sim +70\ °C$	P.339 to P.366	$T_A = -20\ °C$ to $+70\ °C$
P.339 to P.364	$3.3\ V \pm 5\%$	P.341 to P.366	$3.3\ V \pm 0.165\ V$
P.337 to P.364	(Omit)	P.339 to P.366	("-" is written into the blank spaces on the lines of condition, allowance value, and unit in the tables between these pages.)
P.338	(In the Conditions column of item C1) V_{DD}, PVdd = 3.3 V	P.340	(In the Conditions column of item C1) V_{DD}, PV_{DD} = 3.3 V

P.346 — Errors

Item		Symbol	Conditions	Allowable	
				Min.	Max.
Synchronous mode data transfer signal output timing (...					
E17	...	t_{AD}			5
E18	...	t_{AH1}		-5	
E25	...	t_{REDF}			5
E26	...	t_{REDR}			5
E27	...	t_{WEDF}			5
E28	...	t_{WEDR}			5

P.348 — Corrections

Item		Symbol	Conditions	Allowable	
				Min.	Max.
Synchronous mode data transfer signal output timing (...					
E17	...	t_{AD}		$\dfrac{t_{CYC}}{4} \times 4$	5
E18	...	t_{AH1}		-5	10
E25	...	t_{REDF}		-5	5
E26	...	t_{REDR}		-5	5
E27	...	t_{WEDF}		-5	5
E28	...	t_{WEDR}		-5	5

P.350 (Errors)

注) *1

$$m = \begin{cases} n & \text{(multiply by 4)} \\ 2n + 1 & \text{(multiply by 2)} \end{cases} \text{("n" represents the number of wait states.)}$$

P.352 (Corrections)

Note:
m in *1 changes depending on the setting contents of MEMCTR0.

$$m = \begin{cases} n & \text{(FRQ = 1)} \\ 2n + 1 & \text{(FRQ = 0)} \end{cases} \text{("n" represents the number of wait states.)}$$

P.351 (Errors)

(Description of signals in table 13-4-6)

Asynchronous mode data transfer signal output timing (Refer to Fig. 13-4-5.)

P.353 (Corrections)

(Description of signals in table 13-4-6)

DRAM mode data transfer signal output timing (Refer to Fig. 13-4-6.)

P.351 (Errors)

(2) n_R and n_C change according to the DRAMCTR setting. The correspondence is shown below.

$$n_R = 2 + 2\,RTC$$

$$n_C = 4 + \frac{4}{n_f} WC$$

Furthermore, when FRQS = L level, $n_f = 2$, and when FRQS = H level, $n_f = 4$.

P.353 (Corrections)

(2) n_R and n_C change according to the DRAMCTR and MEMCTR0 setting. The correspondence is shown below.

$$n_R = 2 + 2\,RTC$$

$$n_C = 4 + \frac{4}{n_f} WC$$

When FRQ = 0, $n_f = 2$, and when FRQ = 1, $n_f = 4$.

P.352 (Errors)

(Min. of E59, Read data hold time.)

5

P.354 (Corrections)

(Min. of E59, Read data hold time.)

0

P.356 (Errors)

(2) n_D change according to the DRAMCTR setting. The correspondence is shown below.

$$n_D = 6 + \frac{8}{n_f} RERP$$

Furthermore, when FRQS = L level, $n_f = 2$, and when FRQS = H level, $n_f = 4$.

P.358 (Corrections)

(2) n_D change according to the DRAMCTR and MEMCTR0 setting. The correspondence is shown below.

$$n_D = 6 + \frac{8}{n_f} RERP$$

When FRQ = 0, $n_f = 2$, and when FRQ = 1, $n_f = 4$.

P.359 (Errors)

Note:

*1 t_{CYC} is the SYSCLK cycle time.

When FRQS = L level, $n_f = 2$, and when FRQS = H level, $n_f = 4$.

P.361 (Corrections)

Note:

*1 t_{CYC} is the SYSCLK cycle time.

n_f change according to the MEMCTR0 setting.

When FRQ = 0, $n_f = 2$, and when FRQ = 1, $n_f = 4$.

P.359 Fig. 13-4-11 DMA Request Input Timing (1)

P.361 Fig. 13-4-11 DMA Request Input Timing

P.359

P.361 (Title "Fig. 13-4-15 Timer Counter Input Signal Timing" was deleted.)

P.365

P.367 (Japanese fonts in Fig. 13-4-19 put into English.)

P.368 (The Treatment of DSCLK and DSDAT in Table A-1)

..., recommended value for resister is 1KΩ

P.370 (The Treatment of DSCLK and DSDAT in Table A-1)

..., recommended value for resister is 1 kΩ

P.368

P.370 (In Table A-1, VDD was changed into V_{DD} and VSS was changed into V_{SS}.)

P.370

P.372 (In the Table C-1, the row of address x'2000002X was deleted.)

P.370 to P.374

P.372 to P.376 (Following note is added under the Register Maps.)

Note: Accessing areas that are not mounted is prohibited. Operation is not guaranteed if an area that is not mounted is accessed.

P.375 List of Instructions (Code Length, Number of Cycles)

P.377 List of Instructions (Code Length, Number of Cycles *)

P.375

P.377 (Following annotation is added under the table.)

* The number of execution cycles was calculated under the following conditions.

(1) No pipeline extension

(2) 2 cycles for the instruction fetch, 1 cycle for the load/store

Page	Errors	Page	Corrections

P.378 (Errors)

Instruction			Number of Cycles
Bcc	BEQ		2* / 1
	BNE		2* / 1
	BGT		2* / 1
	BGE		2* / 1
	BLE		2* / 1
	BLT		2* / 1
	BHI		2* / 1
	BCC		2* / 1
	BLS		2* / 1
	BCS		2* / 1
	BVC		3* / 2
	BVS		3* / 2
	BNC		3* / 2
	BNS		3* / 2
	BRA		2*
Lcc	LEQ		1 / 2
	LNE		1 / 2
	LGT		1 / 2
	LGE		1 / 2
	LLE		1 / 2
	LLT		1 / 2
	LHI		1 / 2
	LCC		1 / 2
	LLS		1 / 2
	LCS		1 / 2
	LRA		1

P.380 (Corrections)

Instruction			Number of Cycles
Bcc	BEQ		3 / 1*
	BNE		3 / 1*
	BGT		3 / 1*
	BGE		3 / 1*
	BLE		3 / 1*
	BLT		3 / 1*
	BHI		3 / 1*
	BCC		3 / 1*
	BLS		3 / 1*
	BCS		3 / 1*
	BVC		4 / 2*
	BVS		4 / 2*
	BNC		4 / 2*
	BNS		4 / 2*
	BRA		3
Lcc	LEQ		1 / 2*
	LNE		1 / 2*
	LGT		1 / 2*
	LGE		1 / 2*
	LLE		1 / 2*
	LLT		1 / 2*
	LHI		1 / 2*
	LCC		1 / 2*
	LLS		1 / 2*
	LCS		1 / 2*
	LRA		1

P.378 → **P.380** (Folloing annotation is added under the table.)

* Varies according to the state of the instruction buffer.

P.379 (Errors)

Instruction			Number of Cycles
JMP	JMP		3*
	JMP		2*
	JMP		4*

P.381 (Corrections)

Instruction			Number of Cycles
JMP	JMP		3
	JMP		2
	JMP		4

P.379 (Errors)

Instruction			Number of Cycles
RET	RET		4
			4
			4
			4
			5
			8
			9
			10
			11
			12
RETS	RETS		4

P.381 (Corrections)

Instruction			Number of Cycles
RET	RET		6
			6
			6
			6
			7
			10
			11
			12
			13
			14
RETS	RETS		6

Page	Errors	Page	Corrections
P.381	(The 1st line of main text.) The MN10300 Series ...	P.383	(The 1st line of main text.) The MN1030 Series ...
P.384		P.387	([Programming Cautions] is added for PUTX.)
P.385		P.388	([Programming Cautions] is added for PUTCX.)
P.385		P.389	(Following sentences are added to [Programming Cautions] of GETX.) When "udf15 Dm,Dn" is operated, Dm is ignored. The operations of "udf15 imm8,Dn", "udf15 imm16,Dn" and "udf15 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.386		P.390	(Following sentences are added to [Programming Cautions] of GETCHX.) When "udf12 Dm,Dn" is operated, Dm is ignored. The operations of "udf12 imm8,Dn", "udf12 imm16,Dn" and "udf12 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

Page	Errors	Page	Corrections
P.387		P.391	(Following sentences are added to [Programming Cautions] of GETCLX.) When "udf13 Dm,Dn" is operated, Dm is ignored. The operations of "udf13 imm8,Dn", "udf13 imm16,Dn" and "udf13 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.388		P.392	([Programming Cautions] is added for CLRMAC.)
P.389	([Operation] of MULQ.) The range of significant values for the multiplier that is stored in Dn before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in Dn, the quicker the result of the operation can be derived.	P.393	([Operation] of MULQ.) The range of significant values for <u>the multiplicand</u> that is stored in <u>Dm</u> before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller <u>the absolute value of the contents</u> that are stored in <u>Dm</u>, the quicker the result of the operation can be derived.
P.390	([Operation] of MULQI.) The range of significant values for the multiplier that is stored in Dn before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in Dn, the quicker the result of the operation can be derived.	P.394	([Operation] of MULQI.) The range of significant values for <u>the multiplicand</u> that is stored in <u>imm</u> before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, <u>if the number of imm bits is "16" or less, the operation results will be derived faster.</u>
P.391	([Operation] of MULQU.) The range of significant values for the multiplier that is stored in Dn before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in Dn, the quicker the result of the operation can be derived.	P.395	([Operation] of MULQU.) The range of significant values for <u>the multiplicand</u> that is stored in <u>Dm</u> before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in <u>Dm</u>, the quicker the result of the operation can be derived.
P.392	([Operation] of MULQIU.) The range of significant values for the multiplier that is stored in Dn before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, the smaller the contents that are stored in Dn, the quicker the result of the operation can be derived.	P.396	([Operation] of MULQIU.) The range of significant values for <u>the multiplicand</u> that is stored in <u>Dm</u> before the operation is evaluated (starting point: LSB; evaluation unit: 2 bytes) and the operation is only performed for the range containing these significant values. In short, <u>if the number of imm bits is "16" or less, the operation results will be derived faster.</u>
P.396		P.400	(Following sentences are added to [Programming Cautions] of MACIH.) The operations of "udf30 imm32,Dn" is not assured. In addition, a system error interrupt does not occur in these cases.
P.398		P.402	(Following sentences are added to [Programming Cautions] of MACIB.) The operations of "udf32 imm16,Dn" and "udf32 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.401	*MACHU (Signed half-word data multiply-and-accumulate operation instruction: between registers)*	P.405	*MACHU (Unsigned half-word data multiply-and-accumulate operation instruction: between registers)*
P.402		P.406	(Following sentences are added to [Programming Cautions] of MACIHU.) The operations of "udfu31 imm32,Dn" is not assured. In addition, a system error interrupt does not occur in these cases.
P.403	*MACBU (Signed byte data multiply-and-accumulate operation instruction: between registers)*	P.407	*MACBU (Unsigned byte data multiply-and-accumulate operation instruction: between registers)*
P.404		P.408	(Following sentences are added to [Programming Cautions] of MACIBU.) The operations of "udfu33 imm16,Dn" and "udfu33 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.405		P.409	(Following sentences are added to [Programming Cautions] of SAT16.) The operations of "udf04 imm8,Dn", "udf04 imm16,Dn" and "udf04 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.406		P.410	(Following sentences are added to [Programming Cautions] of SAT24.) The operations of "udf05 imm8,Dn", "udf05 imm16,Dn" and "udf05 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

Page	Errors	Page	Corrections

P.407	[Instruction Format (Macro Name)]
	    MCST32, MCST16, MCST8
		P.411	[Instruction Format (Macro Name)]
			    MCST Dm, Dn
			    MCST imm8, Dn

P.407	(From 1st line of [Operation])
	  In addition, depending on the value of Dm, the following operations are performed.
	(1) When the value of Dm is 32 (0x00000020)
	    :
	(2) When the value of Dm is 16 (0x00000010)
	    :
	(3) When the value of Dm is 8 (0x00000008)
	    :
	(4) When the value of Dm is any other value
		P.411	(From 1st line of [Operation])
			  In addition, depending on the value of Dm or imm8, the following operations are performed.
			(1) When the value of Dm or imm8 is 32 (0x00000020)
			    :
			(2) When the value of Dm or imm8 is 16 (0x00000010)
			    :
			(3) When the value of Dm or imm8 is 8 (0x00000008)
			    :
			(4) When the value of Dm or imm8 is any other value

P.408		P.412	(Following sentences are added to [Programming Cautions] of MCST.) The operations of "udf02 imm16,Dn" and "udf02 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

P.409	[Instruction Format (Macro Name)]
	    MCST9 Dn, Dn
		P.413	[Instruction Format (Macro Name)]
			    MCST9 Dn

P.409		P.413	(Following sentences are added to [Programming Cautions] of MCST9.) When "udf03 Dm,Dn" is operated, Dm is ignored. The operations of "udf03 imm8,Dn", "udf03 imm16,Dn" and "udf03 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

P.410	[Instruction Format (Macro Name)]
	    MCST48 Dn, Dn
		P.414	[Instruction Format (Macro Name)]
			    MCST48 Dn

P.410		P.414	(Following sentences are added to [Programming Cautions] of MCST48.) When "udf06 Dm,Dn" is operated, Dm is ignored. The operations of "udf06 imm8,Dn", "udf06 imm16,Dn" and "udf06 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.411		P.415	(Following sentences are added to [Programming Cautions] of BSCH.) The operations of "udf07 imm8,Dn", "udf07 imm16,Dn" and "udf07 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.412	(From 3rd line to 6th line of SWAP's [Operation].) As a result, bits 32 through 24 of Dm are ... are stored in bits 32 through 24 in Dn.	P.416	(From 3rd line to 6th line of SWAP's [Operation].) As a result, bits 31 through 24 of Dm are ... are stored in bits 31 through 24 in Dn.
P.412		P.416	(Following sentences are added to [Programming Cautions] of SWAP.) The operations of "udf08 imm8,Dn", "udf08 imm16,Dn" and "udf08 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.
P.413	(The 1st line of SWAPH's [Operation].) ..., and bits 32 through 24 with	P.417	(The 1st line of SWAPH's [Operation].) ..., and bits 31 through 24 with ....
P.413		P.417	(Following sentences are added to [Programming Cautions] of SWAPH.) The operations of "udf09 imm8,Dn", "udf09 imm16,Dn" and "udf09 imm32,Dn" are not assured. In addition, a system error interrupt does not occur in these cases.

Page	Errors	Page	Corrections

P.414 / P.418 (Errors table)

Preceding instruction	Following instruction	Placement relationship	Notes
Word/half-word data *1 Multiply-and-accumulate instruction	Quick multiplication instruction *3	Following	Insert at least one cycle between the instructions
Word/half-word data *1 Multiply-and-accumulate instruction	Multiply-and-accumulate instruction *4	Following	Insert at least two cycles between the instructions
Word/half-word data *1 Multiply-and-accumulate instruction	MCRH, MCRL access instruction *5	Following	Insert at least three cycles between the instructions
Byte data *2 Multiply-and-accumulate instruction	Multiply-and-accumulate instruction *4	Following	Insert at least one cycle between the instructions
Byte data *2 Multiply-and-accumulate instruction	MCRH, MCRL access instruction *5	Following	Insert at least two cycles between the instructions

P.418 (Corrections table)

Preceding instruction	Following instruction	Placement relationship	Notes
Word/half-word data *1 Multiply-and-accumulate instruction	Quick multiplication instruction *3	Following	Insert at least one cycle between the instructions
Word/half-word data *1 Multiply-and-accumulate instruction	Multiply-and-accumulate instruction *4	Following	Insert at least two cycles between the instructions
Word/half-word data *1 Multiply-and-accumulate instruction	MCRH, MCRL access instruction *5	Following	Insert at least three cycles between the instructions
Byte data *2 Multiply-and-accumulate instruction	Multiply-and-accumulate instruction *4	Following	Insert at least one cycle between the instructions
Byte data *2 Multiply-and-accumulate instruction	MCRH, MCRL access instruction *5	Following	Insert at least two cycles between the instructions
mov [regs], SP instruction	Extension Operation instruction	Following	Insert at least one cycle *6 between the instructions

Page	Errors	Page	Corrections
P.414	(None)	P.418	(The following note is added.) *6: This problem can be avoided by an assembler after V3.1R9. At assembling, nop instruction is automatically inserted between the mov and extended operation instructions.
P.421	The package outline and dimensions of the MN103002A are shown below.	P.425	Fig.F-1 shows the package outline and dimensions of the MN103002A, and Fig.F-2 shows the package outline and dimensions of the MN103002AYB.
P.421	Fig. F-1 Package Outline and Dimensions	P.425	Fig. F-1 Package Outline and Dimensions of MN103002A
-		P.426	("Fig. F-2 Package Outline and Dimensions of MN103002AYB" is added.)
-		-	(In addition to these corrections, how to describe the unit is changed , but the data are not changed.) Example) In the section "3.3 Block Diagram" Error : 13.0M ~ 16.6MHz Correction : 13.0 MHz to 16.6 MHz)
Cover	**MICROCOMPUTER MN10300**	Cover	**MICROCOMPUTER MN1030**
Special attention and precautions	(In the number (1)) "Foreign Exchange and Foreign Trade Control Law"	Special attention and precautions	(In the number (1)) "Foreign Exchange and Foreign Trade Law"
Special attention and precautions	If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book or Matsushita Electronics Corporation's Sales Department.	Special attention and precautions	If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.
Colophon	Issued by Matsushita Electric Industrial Co., Ltd. Matsushita Electronics Corporation © Matsushita Electric Industrial Co., Ltd. © Matsushita Electronics Corporation	Colophon	Issued by Matsushita Electric Industrial Co., Ltd. © Matsushita Electric Industrial Co., Ltd.

Page							420	

The left column of the outer table reads "Page" / "Corrections" for the page 420 section, and "Page" / "Corrections" for the page 424 section.

Semiconductor Company, Matsushita Electric Industrial Co., Ltd.
Nagaokakyo, Kyoto 617-8520, Japan
Tel: (075) 951-8151
http://www.panasonic.co.jp/semicon/
SALES OFFICES

■ **NORTH AMERICA**
●**U.S.A. Sales Office:**
Panasonic Industrial Company [PIC]
• **New Jersey Office:**
Two Panasonic Way Secaucus, New Jersey 07094 U.S.A.
Tel: 1-201-348-5257 Fax:1-201-392-4652
• **Chicago Office:**
1707 N. Randall Road Elgin, Illinois 60123-7847 U.S.A.
Tel: 1-847-468-5720 Fax:1-847-468-5725
• **Milpitas Office:**
1600 McCandless Drive Milpitas, California 95035 U.S.A.
Tel: 1-408-942-2912 Fax:1-408-946-9063
• **Atlanta Office:**
1225 Northbrook Parkway Suite 1-151 Suwanee, GA 30024 U.S.A.
Tel: 1-770-338-6953 Fax:1-770-338-6849
• **San Diego Office:**
9444 Balboa Avenue, Suite 185, San Diego, California 92123 U.S.A.
Tel: 1-619-503-2903 Fax:1-858-715-5545
●**Canada Sales Office:**
Panasonic Canada Inc. [PCI]
5770 Ambler Drive 27 Mississauga, Ontario, L4W 2T3 CANADA
Tel: 1-905-238-2101 Fax:1-905-238-2414

■ **LATIN AMERICA**
●**Mexico Sales Office:**
Panasonic de Mexico, S.A. de C.V. [PANAMEX]
Amores 1120 Col. Del Valle Delegacion Benito Juarez C.P. 03100 Mexico, D.F. MEXICO
Tel: 52-5-488-1000 Fax:52-5-488-1073
• **Guadalajara Office:**
SUCURSAL GUADALAJARA
Av. Lazaro Cardenas 2305 Local G-102 Plaza Comercial Abastos; Col. Las Torres Guadalajara, Jal. 44920 MEXICO
Tel: 52-3-671-1205 Fax:52-3-671-1256
●**Brazil Sales Office:**
Panasonic do Brasil Ltda. [PANABRAS]
Caixa Postal 1641, Sao Jose dos Campos, Estado de Sao Paulo
Tel: 55-12-335-9000 Fax:55-12-331-3789

■ **EUROPE**
●**U.K. Sales Office:**
Panasonic Industrial Europe Ltd. [PIEL]
Willoughby Road, Bracknell, Berks., RG12 8FP, THE UNITED KINGDOM
Tel: 44-1344-85-3671 Fax:44-1344-85-3853
●**Germany Sales Office:**
Panasonic Industrial Europe GmbH [PIEG]
Hans-Pinsel-Strasse 2 85540 Haar, GERMANY
Tel: 49-89-46159-119 Fax:49-89-46159-195

■ **ASIA**
●**Singapore Sales Office:**
Panasonic Semiconductor of South Asia [PSSA]
300 Beach Road, #16-01, The Concourse, Singapore 199555 THE REPUBLIC OF SINGAPORE
Tel: 65-390-3688 Fax:65-390-3689
●**Malaysia Sales Office:**
Panasonic Industrial Company (M) Sdn. Bhd. [PICM]
• **Head Office:**
Tingkat 16B, Menara PKNS Petaling Jaya, No.17, Jalan Yong Shook Lin 46050 Petaling Jaya, Selangor Darul Ehsan, MALAYSIA
Tel: 60-3-7951-6601 Fax:60-3-7954-5968

• **Penang Office:**
Suite 20-07,20th Floor, MWE Plaza, No.8, Lebuh Farquhar,10200 Penang, MALAYSIA
Tel: 60-4-201-5113 Fax:60-4-261-9989
• **Johore Sales Office:**
Menara Pelangi, Suite8.3A, Level8, No.2, Jalan Kuning Taman Pelangi, 80400 Johor Bahru, Johor, MALAYSIA
Tel: 60-7-331-3822 Fax:60-7-355-3996
●**Thailand Sales Office:**
Panasonic Industrial (THAILAND) Ltd. [PICT]
252-133 Muang Thai-Phatra Complex Building, 31st Fl. Rachadaphisek Rd., Huaykwang, Bangkok 10320, THAILAND
Tel: 66-2-693-3428 Fax:66-2-693-3422
●**Philippines Sales Office:** [PISP]
Panasonic Indsutrial Sales Philippines Division of Matsushita Electric Philippines Corporation
102 Laguna Boulevard,Bo.Don Jose Laguna Technopark, Santa. Rosa, Laguna 4026 PHILIPPINES
Tel: 63-2-520-8615 Fax:63-2-520-8629
●**India Sales Office:**
National Panasonic India Ltd. [NPI]
E Block, 510, International Trade Tower Nehru Place, New Delhi_110019 INDIA
Tel: 91-11-629-2870 Fax:91-11-629-2877
●**Indonesia Sales Office:**
P.T.MET & Gobel [M&G]
JL. Dewi Sartika (Cawang 2) Jakarta 13630, INDONESIA
Tel: 62-21-801-5666 Fax:62-21-801-5675
●**China Sales Office:**
Panasonic Industrial (Shanghai) Co., Ltd. [PI(SH)]
Floor 6, Zhong Bao Mansion, 166 East Road Lujian Zui, PU Dong New District, Shanghai, 200120 CHINA
Tel: 86-21-5866-6114 Fax:86-21-5866-8000
Panasonic Industrial (Tianjin) Co., Ltd. [PI(TJ)]
Room No.1001, Tianjin International Building 75, Nanjin Road, Tianjin 300050, CHINA
Tel: 86-22-2313-9771 Fax:86-22-2313-9770
Panasonic SH Industrial Sales (Shenzhen) Co., Ltd.
[PSI(SZ)]
7A-107, International Bussiness & Exhibition Centre, Futian Free Trade Zone, Shenzhen 518048, CHINA
Tel: 86-755-359-8500 Fax:86-755-359-8516
Panasonic Shun Hing Industrial Sales (Hong Kong) Co., Ltd. [PSI(HK)]
11th Floor, Great Eagle Center 23 Harbour Road, Wanchai, HONG KONG
Tel: 852-2529-7322 Fax:852-2865-3697
●**Taiwan Sales Office:**
Panasonic Industrial Sales (Taiwan) Co.,Ltd. [PIST]
• **Head Office:**
6F, 550, Sec. 4, Chung Hsiao E. RD. Taipei, 110, TAIWAN
Tel: 886-2-2757-1900 Fax:886-2-2757-1906
• **Kaohsiung Office:**
6th Floor, Hsin Kong Bldg. No.251, Chi Hsien 1st Road Kaohsiung 800, TAIWAN
Tel: 886-7-346-3815 Fax:886-7-236-8362
●**Korea Sales Office:**
Panasonic Industrial Korea Co., Ltd. [PIKL]
Kukje Center Bldg. 11th Fl., 191 Hangangro 2ga, Youngsan-ku, Seoul 140-702, KOREA
Tel: 82-2-795-9600 Fax:82-2-795-1542